Information Sciences 453 (2018) 323–345

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A new paradigm for uncertain knowledge representation by Plausible Petri nets

Manuel Chiachío*, Juan Chiachío, Darren Prescott, John Andrews

Resilience Engineering Research Group, Faculty of Engineering, University of Nottingham, University Park, Nottingham NG7 2RD, UK

ARTICLE INFO

Article history:
Received 18 October 2016
Revised 31 March 2018
Accepted 7 April 2018
Available online 10 April 2018

Keywords:
Petri nets
Information theory
Knowledge representation
Expert systems

ABSTRACT

This paper presents a new model for Petri nets (PNs) which combines PN principles with the foundations of information theory for uncertain knowledge representation. The resulting framework has been named *Plausible Petri nets* (PPNs). The main feature of PPNs resides in their efficiency to jointly consider the evolution of a discrete event system together with uncertain information about the system state using *states of information*. The paper overviews relevant concepts of information theory and uncertainty representation, and presents an algebraic method to formally consider the evolution of uncertain state variables within the PN dynamics. To illustrate some of the real-world challenges relating to uncertainty that can be handled using a PPN, an example of an expert system is provided, demonstrating how condition monitoring data and expert opinion can be modelled.

1. Introduction

Plausible reasoning is a fundamental human capability that involves the manipulation of perceptions, signs, and information from uncertain surroundings, which allows us to render an uncertain knowledge representation of reality. For many years, uncertain knowledge representation has attracted considerable attention from a large number of researchers in the artificial intelligence (AI) community in order to make real-world knowledge suitable for processing by computers. For decades, Bayesian networks [1] have played a leading role in plausible reasoning and knowledge representation in the AI community [2]. Other formalisms were also used to enable reasoning under uncertainty, such as fuzzy conceptual graphs, influence diagrams, Markov models, neural networks, and dynamic uncertainty causality graphs, to name but a few. See [3] for a recent overview and [4] for a discussion of these methods. However, despite the efficiency that these methods have demonstrated for uncertain knowledge representation and reasoning at a system level, their pragmatism is not sufficient when features of real-life systems like synchrony, concurrency, and complexity are considered in the modelling. To efficiently approach these modelling issues, an increasing number of researchers reported progress towards the modelling of knowledge-based reasoning using Petri nets (PNs) [5]. PNs are bipartite directed graphs (digraphs), which are used to model and analyse discrete event systems. The basic concepts relative to the theory of PNs are summarised in [6], whereas a recent broad historical perspective on this field is provided in [7]. A tutorial for practical engineering applications of PNs can be found in [8]. PNs are well-suited to modelling knowledge representation at a system level since they provide a graphical

* Corresponding author.
E-mail address: manuel.chiachio-ruano1@nottingham.ac.uk (M. Chiachío).

https://doi.org/10.1016/j.ins.2018.04.029

support for interpretation, but also because they rely on mathematical principles which enable their implementation and simulation in a rational manner. However, the main drawback of PNs is that they are not adequate to modelling uncertainty when considered under the original definition proposed by Carl A. Petri [5], since their dynamics are based on sequences of Boolean operations. Thus, over the last three decades, researchers developed variants of the original PNs in order to handle uncertain information. Of the many PN variants, Fuzzy Petri nets (FPNs) have received much attention, since they are of particular interest for tackling fuzziness in systems modelling [9,10]. Following the pioneering FPN work in [11], a number of contributions have been devoted to enhancing FPNs with improved rules of inference and uncertainty management, e.g. [12,13]. See [14,15] for a discussion of the various FPN improvements, and [16,17] for recent literature reviews on this topic. Moreover, other approaches appeared to consider not only fuzziness over the PN structure but also other types of uncertainty like, for example, Possibilistic PNs [18,19].

None of the PN variants developed to handle uncertainty are well-suited to embedding plausible reasoning into the PN formalism, nor do they consider the hybrid nature of real-world dynamical systems, consisting of a combination of discrete and continuous processes whose evolution may be uncertain. In this context, this paper proposes a new class of models within the PN paradigm, which has been developed by combining information theory principles with the PN technique. The resulting framework has been named *Plausible Petri Nets* (PPNs). The pivotal idea behind PPNsis to consider two interacting subnets: (1) the *symbolic subnet*, where the *tokens* are objects in the sense of integer moving units, as in classical PNs [5]; (2) the *numerical subnet*, where tokens are density functions (also referred to here as *states of information*) about a state variable. In PPNs, the uncertainty is accounted for through the states of information, which provide a mapping that assigns to each possible numerical value of the state variable its relative plausibility [20,21]. The resulting model is hybrid since its response is determined by the interaction between the continuous and discrete dynamics.

In the literature, first attempts to represent system hybrid characteristics using PNs were proposed by [22,23] (revisited afterwards in [24] and [25], respectively). Hybrid Petri nets (HPNs) still constitute an active area of research today, since they provide a satisfactory solution to the *state explosion problem* typically present in the PN-based analysis of engineering systems [26]. Several formalisms falling in this category have been recently presented in the literature, mostly motivated by particular engineering applications, e.g. see [27–29]. However, HPNs suffer from some restrictions for uncertain knowledge representation since: (1) uncertainty cannot be fully accounted for, despite some approaches like [30,31], which make a partial attempt to model uncertainty by considering stochastic firing of transitions or by adding randomness to the numerical subnet; (2) the numerical and symbolic subnets interact in a unidirectional manner (namely, the symbolic subnet may influence the numerical subnet behaviour, whilst the opposite never occurs [26]), which ultimately limits the knowledge representation and reasoning capabilities. It is worth mentioning that a particular approach within the HPN paradigm, referred to as *Hybrid Particle Petri nets* (HPPNs), was first presented in [32] and further developed in [33,34] to capture the hybrid aspects of the system dynamics and also the uncertainties. Nonetheless, their formulations are based on a discretisation of the numerical values through a limited set of *weighted particles* acting as moving tokens within the numerical subnet, which limits their ability to represent the uncertainty.

With the proposed PPNs, the uncertainty is rigorously accounted for through adaptive states of information about state (numerical) variables defining a flow of uncertain information over time of execution. Hence, the net dynamics is not only associated with the concept of adding and subtracting integer values (classical tokens) in congruence with the state of the numerical subnet, but also with the operation of adding and subtracting states of information about the system state, which represents a novelty with respect to previous HPN approaches reported in the literature. Moreover, the numerical and symbolic subnets in PPNs interact in a bidirectional manner, which overcomes the aforementioned limitation of most of the HPN approaches.

Apart from introducing the concept of PPNs for uncertain knowledge representation, this paper also

1. formally defines the PPN execution rules;
2. analyses structural properties from an information theoretic approach;
3. proposes a particle-based approximation for the states of information to confer versatility to the PPNs;
4. provides a pseudocode implementation for PPNs;
5. illustrates the applicability of PPNs through a numerical example about an expert system.

Initial ideas about the PPN methodology have been first presented in a conference paper [35] but here we provide fuller explanation, insightful examples, and extended results. The remainder of the paper is organised as follows. Section 2 briefly overviews basic concepts about PNs and plausible reasoning before introducing the mathematical basis of PPNs. The general formalism for the proposed PPNs and their execution rules, are mathematically described in Section 3. In Section 4, an algorithmic description of PPNs is provided. Section 5 illustrates our approach through a fault tolerant expert system and Section 6 discusses the main findings and provides a comparative analysis with different PN paradigms. Finally, Section 7 gives concluding remarks.

2. Basic concepts

2.1. Petri nets

PNs were introduced in the thesis dissertation *Kommunikation mit Automaten* by Carl Petri in 1962 [5]. From a mathematical perspective, PNs are bipartite directed graphs (digraphs) which are broadly used for modelling the dynamics of systems.

Definition 1 (Petri net [6])**.** A PN is defined as a tuple $\mathfrak{N} = \langle \mathbf{P}, \mathbf{T}, \mathbf{F}, \mathbf{W}, \mathbf{M}_0 \rangle$, where:

1. $\mathbf{P} = \{p_1, p_2, \ldots, p_{n_p}\}$ is a finite set of places;
2. $\mathbf{T} = \{t_1, t_2, \ldots, t_{n_t}\}$ is a finite set of transitions, such that $\mathbf{T} \cap \mathbf{P} = \emptyset$ and $\mathbf{T} \cup \mathbf{P} = \emptyset$;
3. $\mathbf{F} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$ represents a set of directed arcs connecting places to transitions and vice versa;
4. $\mathbf{W} : \mathbf{F} \to \mathbb{N}_{>0}$ is a weight function, which assigns a value (1 by default) to each arc within $\mathbf{F}$;
5. $\mathbf{M}_0 : \mathbf{P} \to \mathbb{N}$ is the initial marking, which expresses the initial distribution of tokens over the set of places.

The following notation will also be considered:

${}^\bullet t$ is the set of input places of transition t, also referred to as the *pre-set* of t;
$t^\bullet$ is the set of output places of transition t, also referred to as the *post-set* of t;

The dynamics of a PN can be described through a state equation defined as follows [6]:

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{A}^T \mathbf{u}_k \tag{1}$$

where $k \in \mathbb{N}$ is the time index, and $\mathbf{A}$ is an $n_t \times n_p$ matrix typically referred to as the *incidence matrix*, which can be obtained as the result of subtracting the *backward incidence matrix* ($\mathbf{A}^-$) from the *forward incidence matrix* ($\mathbf{A}^+$), i.e.:

$$\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^- \tag{2}$$

where $\mathbf{A}^+ = \left[a_{ij}^+\right]$, $\mathbf{A}^- = \left[a_{ij}^-\right]$, $i = 1, \ldots, n_t$, $j = 1, \ldots, n_p$. The element a_{ij}^+ is the weight of the arc from transition $t_i \in \mathbf{T}$ to output place $p_j \in \mathbf{P}$, whereas a_{ij}^- is the weight of the arc to transition t_i from input place p_j. The term $\mathbf{u}_k = (u_{1,k}, u_{2,k}, \ldots, u_{n_t,k})^T$ is the *firing vector*, a vector of binary values whose ith component takes 1 if transition t_i is fired, and 0 otherwise. A necessary condition for t_i to be fired is to have been enabled, which occurs if each input place of t_i is marked with at least a_{ij}^- tokens. Mathematically:

$$M(j) \geqslant a_{ij}^- \quad \forall p_j \in {}^\bullet t_i \tag{3}$$

where $M(j) \in \mathbb{N}$ is the marking for place p_j.

In practical applications, transitions are typically assigned with time delays which are useful for performance evaluation and scheduling problems of dynamical systems [6]. The resulting PNs are called *Timed Petri nets* if the delays are deterministic, and *Stochastic Petri nets* if the delays are randomly chosen by sampling distributions [36]. In such cases, a transition is time-enabled once its time delay has passed provided that the condition given by Eq. (3) is satisfied. If more than one timed transition is enabled, then an execution policy can be used to specify the procedure whereby the transition to fire is chosen [6]. Typically, the execution policy is specified by the transition with the minimum time delay (race policy), although other execution policies can also be applied [37].

2.2. Interpretation of uncertainty

Consider a measurable space $\mathcal{X} \subset \mathbb{R}^d$, where $\mathcal{A} \subseteq \mathcal{X}$ is a subspace representing a certain event or proposition over $\mathcal{X}$. Most readers will be familiar with the concept of probability $P(\mathcal{A})$ as a Lebesgue measure of $\mathcal{A}$, i.e.:

$$P(\mathcal{A}) = \int_{\mathcal{A}} f(\mathbf{x}) d\mathbf{x} \geqslant 0 \tag{4}$$

where $f(\mathbf{x})$ is a *density function*, a Lebesgue integrable function that can be normalised such that $\int_{\mathcal{X}} f(\mathbf{x}) d\mathbf{x} = 1$, which is equivalent to saying that $P(\mathcal{X}) = 1$. In this work, we adopt a subjective interpretation of probability as epistemic uncertainty whereby $f(\mathbf{x})$ represents the *degree of belief* of the various possible (plausible) values of the uncertain variable $\mathbf{x} \in \mathcal{X}$. This interpretation is not well known in the engineering community, where there is a widespread belief that probability only applies to aleatory uncertainty (inherent randomness) and not to epistemic uncertainty (incomplete information) [38]. Examples of this epistemic uncertainty could arise in the case of the measurement of a state variable given by a noisy sensor, or the consideration of several expert opinions on the value of such variable. Henceforth, under a plausibility-based interpretation of uncertainty, probability cannot be viewed as an absolute measure of an event being true or false, but as a multi-valued logic which expresses the *relative plausibility* of several possibilities [39].

Definition 2 (State of information [20])**.** The degree of belief about the values of the variable $\mathbf{x} \in \mathcal{X}$ is referred to as the state of information of $\mathbf{x}$, which is represented using the density function $f(\mathbf{x})$.

In this sense, $P(\mathcal{A})$ in Eq. (4) is interpreted as the plausibility of the set of possible values $\mathbf{x} \in \mathcal{A} \subseteq \mathcal{X}$ given a state of information about them provided by $f(\mathbf{x})$. From Definition 2, the *homogeneous density function* $\mu(\mathbf{x}) \neq 0$ can be defined as a particular density function to represent the state of complete ignorance about $\mathbf{x} \in \mathcal{X}$, henceforth providing a reference probability model for $\mathbf{x}$ in the absence of any other information [20,40]. In case that $\mathcal{X}$ is a linear space, it is demonstrated that $\mu(\mathbf{x}) = cte$, i.e. $\mu(\mathbf{x})$ represents a uniform density function over $\mathcal{X}$.

Suppose now that two states of information are available about the same variable $\mathbf{x}$, namely $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$, which we may want to combine to obtain a single state of information. This can be achieved by extending the logic operators AND ($\wedge$) and OR ($\vee$) from Boolean logic for logical propositions to the *conjunction* and *disjunction* of states of information, as follows.

Definition 3 (Conjunction of states of information [20])**.** Let $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$ be two states of information about variable $\mathbf{x} \in \mathcal{X}$. The density $(f_a \wedge f_b)(\mathbf{x})$ represents the conjunction of states of information given by $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$, which can be obtained as [20]:

$$\left(f_a \wedge f_b\right)(\mathbf{x}) = \frac{1}{\alpha} \frac{f_a(\mathbf{x}) f_b(\mathbf{x})}{\mu(\mathbf{x})} \tag{5}$$

In the last equation, $\mu(\mathbf{x})$ is the homogeneous density function, and α a normalising constant.

Note that there might be cases where the calculation of the normalising constant α in Eq. (5) involves an intractable integral. Furthermore, in practical engineering applications, the density functions $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$ may be defined through samples. Hence, sampling-based algorithms (e.g. particle methods) [41] can be used in these cases to circumvent the evaluation of the normalising constant with a feasible computational cost. In particle methods, a set of N samples $\left\{\mathbf{x}^{(n)}\right\}_{n=1}^{N}$ with associated "weights" (relative likelihoods) $\left\{\omega^{(n)}\right\}_{n=1}^{N}$ are used to obtain an approximation of the required density function [e.g. $(f_a \wedge f_b)(\mathbf{x})$], as follows:

$$\left(f_a \wedge f_b\right)(\mathbf{x}) \approx \sum_{n=1}^{N} \omega^{(n)} \delta\left(\mathbf{x} - \mathbf{x}^{(n)}\right) \tag{6}$$

where $\mathbf{x}^{(n)}$ denotes the nth sample obtained from $(f_a \wedge f_b)(\mathbf{x})$, δ is the Dirac delta, and $\omega^{(n)}$ is the weight of $\mathbf{x}^{(n)}$, which can be obtained for the case of $\mathcal{X}$ being a linear space as follows [42]:

$$\omega^{(n)} = \frac{f_a(\mathbf{x}^{(n)}) f_b(\mathbf{x}^{(n)})}{\sum_{n=1}^{N} f_a(\mathbf{x}^{(n)}) f_b(\mathbf{x}^{(n)})} \tag{7}$$

A pseudocode implementation to obtain particles from the conjunction $(f_a \wedge f_b)(\mathbf{x})$ is provided as Algorithm 1.

Algorithm 1 Particle approximation of conjunction of states of information. Ref. [43] is cited below.

Inputs: N, $f_a(\mathbf{x})$, $f_b(\mathbf{x})$ ▷ {number of particles and states of information}

Outputs: $\left\{\mathbf{x}^{(n)}, \omega^{(n)}\right\}_{n=1}^{N}$, where $\mathbf{x}^{(n)} \sim (f_a \wedge f_b)(\mathbf{x})$

Begin

1: Sample $\left\{\left(\tilde{\mathbf{x}}_a^{(n)}, \tilde{\omega}_a^{(n)}\right)\right\}_{n=1}^{N}$ from $f_a(\mathbf{x})$ ▷ {e.g. use rejection sampling [43]}

2: Set $\mathbf{x}^{(n)} \leftarrow \tilde{\mathbf{x}}_a^{(n)}$, $n = 1, \ldots, N$

3: Obtain $\hat{\omega}^{(n)} \leftarrow f_b\left(\mathbf{x}^{(n)}\right)$, $n = 1, \ldots, N$ ▷ {unnormalised weights}

4: Normalise weights $\omega^{(n)} \leftarrow \frac{\hat{\omega}^{(n)}}{\sum_{n=1}^{N} \hat{\omega}^{(n)}}$, $n = 1, \ldots, N$

Definition 4 (Disjunction of states of information [20])**.** Let $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$ be two states of information about variable $\mathbf{x} \in \mathcal{X}$. $(f_a \vee f_b)(\mathbf{x})$ represents the disjunction of states of information given by $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$, which can be expressed as a normalised density function [20]:

$$\left(f_a \vee f_b\right)(\mathbf{x}) = \frac{1}{\beta}\left(f_a(\mathbf{x}) + f_b(\mathbf{x})\right) \tag{8}$$

where β is a normalising constant.

A pseudocode implementation for the disjunction of states of information is provided using samples from $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$, and referred to as Algorithm 2. Note that the disjunction operation can be easily extended to the case of multiple states of information (e.g. $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})$), as follows [40]:

$$\left(f_1 \vee \cdots \vee f_m\right)(\mathbf{x}) = \frac{1}{\beta} \sum_{i=1}^{m} f_i(\mathbf{x}) \tag{9}$$

Algorithm 2 Particle approximation of disjunction of states of information. Ref. [43] is cited below.

Inputs: N, $f_a(\mathbf{x})$, $f_b(\mathbf{x})$ ▷ {number of particles and states of information}

Outputs: $\left\{\mathbf{x}^{(n)}, \omega^{(n)}\right\}_{n=1}^{N}$, where $\mathbf{x}^{(n)} \sim (f_a \vee f_b)(\mathbf{x})$

Begin

1: Sample $\left\{\left(\tilde{\mathbf{x}}_a^{(n)}, \tilde{\omega}_a^{(n)}\right)\right\}_{n=1}^{N}$ from $f_a(\mathbf{x})$ ▷ {e.g. use rejection sampling [43]}

2: Sample $\left\{\left(\tilde{\mathbf{x}}_b^{(n)}, \tilde{\omega}_b^{(n)}\right)\right\}_{n=1}^{N}$ from $f_b(\mathbf{x})$

3: $\left\{1/2\,\tilde{\omega}_a^{(n)} \leftarrow \tilde{\omega}_a^{(n)}\right\}_{n=1}^{N}$ ▷ {Modify particle weights}

4: $\left\{1/2\,\tilde{\omega}_b^{(n)} \leftarrow \tilde{\omega}_b^{(n)}\right\}_{n=1}^{N}$ ▷ {Modify particle weights}

5: Concatenate $\left\{\left(\tilde{\mathbf{x}}_a^{(n)}, \tilde{\mathbf{x}}_b^{(n)}\right)\right\}_{n=1}^{N}$ and renumber as $\left(\tilde{\mathbf{x}}^{(1)}, \tilde{\mathbf{x}}^{(2)}, \ldots, \tilde{\mathbf{x}}^{(2N)}\right)$

6: Concatenate $\left\{\left(\tilde{\omega}_a^{(n)}, \tilde{\omega}_b^{(n)}\right)\right\}_{n=1}^{N}$ and renumber as $\left(\tilde{\omega}^{(1)}, \tilde{\omega}^{(2)}, \ldots, \tilde{\omega}^{(2N)}\right)$

7: $\left\{\mathbf{x}^{(n)}, \omega^{(n)}\right\}_{n=1}^{N} \leftarrow \texttt{resampling}\left\{\tilde{\mathbf{x}}^{(n)}, \tilde{\omega}^{(n)}\right\}_{n=1}^{2N}$

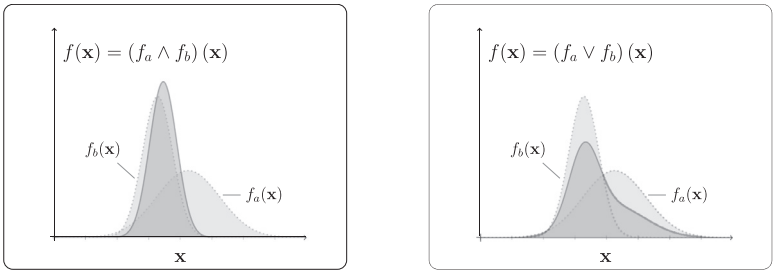


Fig. 1. Illustrative example of the conjunction (left) and disjunction (right) of states of information $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$.

which can be straightforwardly approximated using particles through Algorithm 2 by aggregating samples from $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})$ and considering $1/m$ as the modifying constant for particle weights in steps 3 and 4. Fig. 1 provides us with a conceptual illustration of the conjunction and disjunction of states of information over some arbitrary densities $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$. From left to right, the panels show the resulting density $f(\mathbf{x})$ from the conjunction and disjunction operation, respectively. In both cases, the result has been represented superimposed over $f_a(\mathbf{x})$ and $f_b(\mathbf{x})$ for clarity.

Definition 5 (Conditional probability density [20,40])**.** Let us consider a state of information $f(\mathbf{x})$ defined over $\mathcal{X}$ along with the proposition $\mathbf{x} \in \mathcal{B}$, where $\mathcal{B} \subseteq \mathcal{X}$ is a region of the $\mathbf{x}$-space. Then, the conditional density function $f(\mathbf{x}|\mathbf{x} \in \mathcal{B})$ can be viewed as a special case of a conjunction of states of information between $f(\mathbf{x})$ and a special density $\mu_{\mathcal{B}}(\mathbf{x})$ defined as:

$$\mu_{\mathcal{B}}(\mathbf{x}) = \begin{cases} k\mu(\mathbf{x}), & \text{if} \quad \mathbf{x} \in \mathcal{B} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $\mu(\mathbf{x})$ is the homogeneous density function, and $k > 0$ is a normalising constant.

A practical instance of the aforementioned definition is given when $\mathcal{B}$ can be defined through a performance function $g : \mathcal{X} \to \mathbb{R}$, as a region of the $\mathbf{x}$-space corresponding to the performance function exceeding some specified threshold level b, i.e. $\mathcal{B} = \left\{\mathbf{x} \in \mathcal{X} : g(x) > b\right\}$. By combining Eqs. (5) and (10), the conjunction of states of information between $f(\mathbf{x})$ and $\mu_{\mathcal{B}}(\mathbf{x})$ can be written as:

$$\begin{aligned} (f \wedge \mu_{\mathcal{B}})(\mathbf{x}) &= k_1 \frac{f(\mathbf{x}) k\mu(\mathbf{x})}{\mu(\mathbf{x})} = k_2 f(\mathbf{x})_{|\mathbf{x}\in\mathcal{B}} \triangleq f(\mathbf{x}|\mathbf{x} \in \mathcal{B}) \\ &= \begin{cases} f(\mathbf{x}), & \text{if} \quad g(\mathbf{x}) > b \\ (\emptyset), & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

where k_1, k_2 are normalising constants. Note that the result of the conjunction $(f \wedge \mu_{\mathcal{B}})(\mathbf{x})$ is equal to $f(\mathbf{x})$ except for the condition of being restricted upon the region $\mathcal{B}$, since the relative information of $f(\mathbf{x})$ with respect to $\mu_{\mathcal{B}}(\mathbf{x})$ is exactly the information content of $f(\mathbf{x})$ within $\mathcal{B}$ [20]. Fig. 2 provides us with an illustrative example of the conditional density function

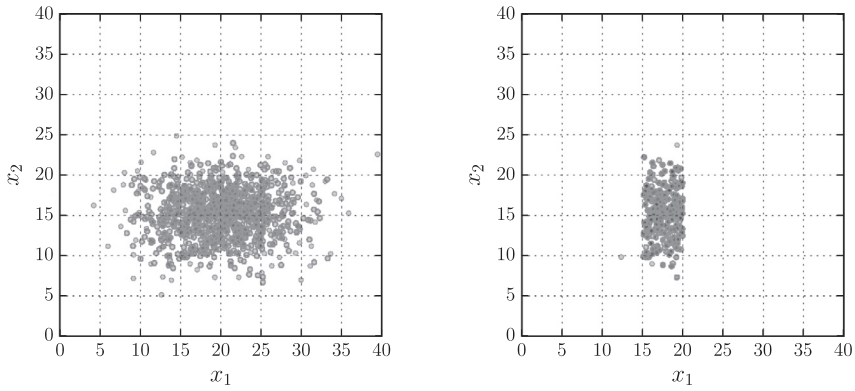


Fig. 2. Illustrative example of a conditional probability density $f(\mathbf{x}|\mathbf{x} \in \mathcal{B})$ (shown in the right panel) obtained as indicated by Eq. (11). Each subplot presents samples (circles) in the state space $\mathcal{X} \subseteq \mathbb{R}^2$.

of a bi-dimensional state variable (right panel) obtained as a conjunction of states of information between $f(\mathbf{x})$ (left panel) and $\mu_{\mathcal{B}}(\mathbf{x})$, where $f(\mathbf{x}) \sim \mathcal{N}\left([20, 15],\ \mathbf{\Sigma}_{p_1}\right)$ being $\mathbf{\Sigma}_{p_1} = diag(5^2, 3^2)$. Here, the region $\mathcal{B}$ is defined as $\mathcal{B} = \left\{\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2 : 15 \leqslant x_1 \leqslant 20\right\}$. In this example, a total of $N = 1000$ samples have been used in Algorithm 1 for the particle approximation of the conjunction of the states of information.

3. Plausible Petri nets

A mathematical description of PPNs is provided below.

3.1. Concept

Definition 6 (Plausible Petri nets)**.** A PPN is defined as a 9-tuple $\mathfrak{M} = \left\langle \mathbf{P}, \mathbf{T}, \mathbf{F}, \mathbf{W}, \mathbf{D}, \mathcal{X}, \mathcal{G}, \mathcal{H}, \mathbf{M}_0 \right\rangle$, where:

1. $\mathbf{P}$ denotes the set of places which is partitioned into numerical places $\mathbf{P}^{(\mathcal{N})} \in \mathbb{N}^{n_p^{(\mathcal{N})}}$, and symbolic places $\mathbf{P}^{(\mathcal{S})} \in \mathbb{N}^{n_p^{(\mathcal{S})}}$, such that $\mathbf{P}^{(\mathcal{N})} \cup \mathbf{P}^{(\mathcal{S})} = \mathbf{P}$, and $\mathbf{P}^{(\mathcal{N})} \cap \mathbf{P}^{(\mathcal{S})} = \emptyset$. Superscripts $n_p^{(\mathcal{N})}$, $n_p^{(\mathcal{S})}$ represent the number of numerical and symbolic places, respectively;
2. $\mathbf{T}$ is the set of transitions which is partitioned into numerical transitions $\mathbf{T}^{(\mathcal{N})} \in \mathbb{N}^{n_t^{(\mathcal{N})}}$ and symbolic transitions $\mathbf{T}^{(\mathcal{S})} \in \mathbb{N}^{n_t^{(\mathcal{S})}}$, where $\mathbf{T}^{(\mathcal{N})} \cup \mathbf{T}^{(\mathcal{S})} = \mathbf{T}$, and $\mathbf{T}^{(\mathcal{N})} \cap \mathbf{T}^{(\mathcal{S})} \neq \emptyset$. Analogously, $n_t^{(\mathcal{N})}$, $n_t^{(\mathcal{S})}$ denote the number of numerical and symbolic transitions, respectively. Those transitions that belong to $\mathbf{T}^{(\mathcal{N})} \cap \mathbf{T}^{(\mathcal{S})}$ are referred to as mixed transitions;
3. $\mathbf{F}$ denotes the set of arcs which contains ordered pairs of nodes that indicate the connections between transitions and places, i.e. $\mathbf{F} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$, hence $\mathbf{F} \subset \mathbb{N}^{n_p^{(\mathcal{N})} + n_p^{(\mathcal{S})}} \times \mathbb{N}^{n_t^{(\mathcal{N})} + n_t^{(\mathcal{S})}}$;
4. $\mathbf{W}$ is a set of non-negative weights applied to each arc within $\mathbf{F}$ (1 by default). The set is partitioned into two subsets, $\mathbf{W}^{(\mathcal{N})}$ and $\mathbf{W}^{(\mathcal{S})}$, each one corresponding to the numerical and symbolic subnet respectively, such that $\mathbf{W}^{(\mathcal{N})} \cup \mathbf{W}^{(\mathcal{S})} = \mathbf{W}$, $\mathbf{W}^{(\mathcal{N})} \cap \mathbf{W}^{(\mathcal{S})} = \emptyset$;
5. $\mathbf{D}$ is a set of switching delays for the symbolic and mixed transitions (0 by default);
6. $\mathcal{X} \subset \mathbb{R}^d$ is the space of the state variable $\mathbf{x}$ and d the dimension of $\mathcal{X}$;
7. $\mathcal{G}$ is a set of density functions associated with the numerical places and transitions;
8. $\mathcal{H}$ is a set of equations representing the dynamics of the state variable $\mathbf{x} \in \mathcal{X}$. They can be of different types (e.g. difference equations, linear, nonlinear, deterministic, etc.) and represent the temporal evolution of the state variable $\mathbf{x}$;
9. $\mathbf{M}_0$ is the initial marking of the net, which is given by the pair of vectors $\mathbf{M}_0^{(\mathcal{N})}$ and $\mathbf{M}_0^{(\mathcal{S})}$ for numerical and symbolic places, respectively.

Observe from Definition 6 that two subnets can be differentiated from the referred tuple $\mathfrak{M}$: (a) the numerical subnet given by the tuple $\left\langle \mathbf{P}^{(\mathcal{N})}, \mathbf{T}^{(\mathcal{N})}, \mathbf{E}, \mathbf{W}^{(\mathcal{N})}, \mathcal{X}, \mathcal{F}, \mathcal{H}, \mathbf{M}_0^{(\mathcal{N})} \right\rangle$, which accounts for the numerical behaviour of the system; and (b) a symbolic subnet defined by $\left\langle \mathbf{P}^{(\mathcal{S})}, \mathbf{T}^{(\mathcal{S})}, \mathbf{E}, \mathbf{W}^{(\mathcal{S})}, \mathbf{D}, \mathbf{M}_0^{(\mathcal{S})} \right\rangle$.

3.2. Modelling assumptions and properties

Let $\mathbf{x}_k \in \mathcal{X}$ represents the state variable of the system at time $k \in \mathbb{N}$. The following are assumptions and interpretations adopted to mathematically describe a PPN model:

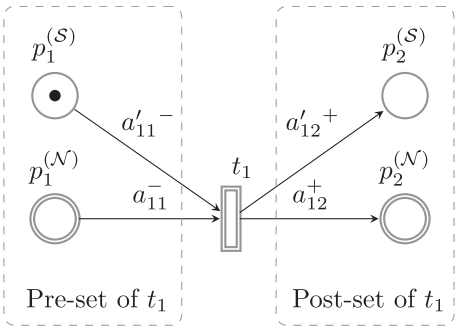


Fig. 3. Illustration of a sample PPNwith two numerical places ($p_1^{(\mathcal{N})}$, $p_2^{(\mathcal{N})}$), two symbolic places ($p_1^{(\mathcal{S})}$, $p_2^{(\mathcal{S})}$), and one transition (t_1). The dashed rectangles are to highlight the pre-set and post-set of transition t_1.

1. As in PNs, symbolic places $\mathbf{P}^{(\mathcal{S})}$ contain tokens which account for the discrete behaviour of the system;
2. A numerical place $p_j^{(\mathcal{N})}$ contains a state of information about $\mathbf{x}_k$ given by $f^{p_j}(\mathbf{x}_k) \in \mathcal{G}$, where f^{p_j} is a density function;
3. Transition $t_i \in \mathbf{T}^{(\mathcal{N})}$ carries information about $\mathbf{x}_k$ through the corresponding state of information given by $f^{t_i}(\mathbf{x}_k) \in \mathcal{G}$;
4. The arc weights $a_{ij}^{+}, a_{ij}^{-} \in \mathbf{W}^{(\mathcal{N})} \subset \mathbb{R}^{+}$ provide us with a measure of the *importance* of the information that flows from/to the corresponding transition. The *incidence matrix* for the numerical subnet is defined from $\mathbf{W}^{(\mathcal{N})}$ as $\mathbf{A}^{(\mathcal{N})} = \left[a_{ij}\right]$, where $a_{ij} = a_{ij}^{+} - a_{ij}^{-},\ i = 1, \ldots, n_t^{(\mathcal{N})},\ j = 1, \ldots, n_p^{(\mathcal{N})}$;
5. Correspondingly, $a_{ij}^{'+}, a_{ij}^{'-} \in \mathbf{W}^{(\mathcal{S})} \subset \mathbb{N}$ denote the arc weights for the symbolic subnet, where the incidence matrix $\mathbf{A}^{(\mathcal{S})} = \left[a'_{ij}\right]$ is obtained as $a'_{ij} = a_{ij}^{'+} - a_{ij}^{'-},\ i = 1, \ldots, n_t^{(\mathcal{S})},\ j = 1, \ldots, n_p^{(\mathcal{S})}$;
6. The marking comprises two column vectors: $\mathbf{M}_k^{(\mathcal{N})}$ for the numerical subnet, and $\mathbf{M}_k^{(\mathcal{S})}$ for the symbolic one, so that $\mathbf{M}_k = \left(\mathbf{M}_k^{(\mathcal{N})}, \mathbf{M}_k^{(\mathcal{S})}\right)$. Mathematically, $\mathbf{M}_k^{(\mathcal{N})}$ is expressed through a column vector of density functions, specified by $\mathbf{M}_k^{(\mathcal{N})} = \left(f_k^{p_1}(\mathbf{x}_k), f_k^{p_2}(\mathbf{x}_k), \ldots, f_k^{p_{n_p^{(\mathcal{N})}}}(\mathbf{x}_k)\right)^T$. Similarly, $\mathbf{M}_k^{(\mathcal{S})}$ is expressed through a column vector of integer values so that its j-th component represents the number of tokens visiting $p_j^{(\mathcal{S})}$ at k;
7. The system evolves in discrete-time with rate $k \in \mathbb{N}$. The system dynamics of both subnets are coupled through *mixed transitions*, whose firing allows us to sequentially activate/deactivate the referred subnets in a synchronised manner.

A PPN model is shown in Fig. 3 for illustration purposes. A double line is adopted to represent the nodes which belong to the numerical subnet, and a single line for the rest.

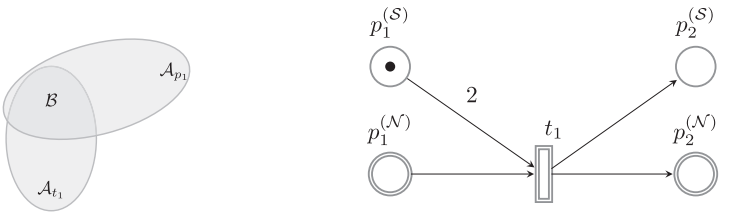
Definition 7 (Transition firing of PPNs)**.** Transition $t_i \in \mathbf{T}$ is fired at discrete time k if:

1. Time delay τ_i has elapsed and each symbolic place from the pre-set of t_i has enough tokens according to their input arc weight (mathematically, $\forall p_j^{(\mathcal{S})} \in {}^{\bullet}t_i,\ M_k^{(\mathcal{S})}(j) \geqslant a_{ij}^{'-}$), for $t_i \in \mathbf{T}^{(\mathcal{S})}$;
2. The conjunction of states of information between $f^{t_i}(\mathbf{x}_k)$ and each of the density functions of the numerical places from the pre-set of t_i, does not yield a null-probability to any subspace $\mathcal{B} \subset \mathcal{X}$. Mathematically, $\forall p_j^{(\mathcal{N})} \in {}^{\bullet}t_i,\ \int_{\mathcal{B}} \left(f^{p_j} \wedge f^{t_i}\right)(\mathbf{x}_k) d\mathbf{x}_k > 0$, which in this case applies for numerical transitions, i.e., $t_i \in \mathbf{T}^{(\mathcal{N})}$;
3. Conditions (1) and (2) are both satisfied when t_i is a mixed transition, i.e. $t_i \in (\mathbf{T}^{(\mathcal{S})} \cap \mathbf{T}^{(\mathcal{N})})$.

Note from Condition (1) that $t_i \in \mathbf{T}^{(\mathcal{S})}$ is timed enabled when time delay τ_i has elapsed, whereas it is logically enabled when the condition $M_k^{(\mathcal{S})}(j) \geqslant a_{ij}^{'-},\ \forall p_j^{(\mathcal{S})} \in {}^{\bullet}t_i$, is met. Once fired, t_i removes $a_{ij}^{'-}$ tokens from each jth pre-set place, as a usual discrete transition. Note also that because symbolic transitions in PPNs are defined as timed, hence execution policies might apply as in Timed Petri nets and Stochastic Petri nets [37]. Their definition and specification within the context of PPNs is out of the scope of this paper. For simplicity but no loss of generality, we assume here that a conflict resolution policy (e.g. *race policy*) applies when more than one timed transition is enabled. An *enabling memory policy* can be used to keep the transition delay values until they become disabled (see further details in [37]).

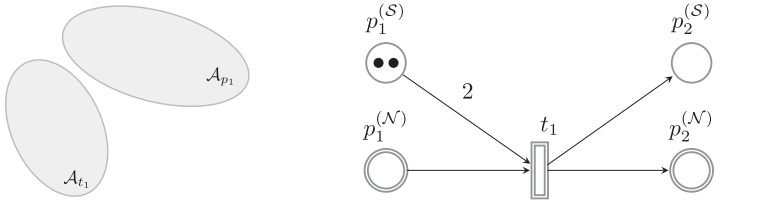
Finally, it is worth noting from Condition 2) that a necessary and sufficient condition for the integral $\int_{\mathcal{B}} \left(f^{p_j} \wedge f^{t_i}\right)(\mathbf{x}_k) d\mathbf{x}_k > 0$ is that $\int_{\mathcal{B}} f^{p_j}(\mathbf{x}_k) d\mathbf{x}_k > 0$ and $\int_{\mathcal{B}} f^{t_i}(\mathbf{x}_k) d\mathbf{x}_k > 0$ [20]

Lemma 1. *Any numerical transition with $\mu(\mathbf{x})$ as associated probability density will always be fired irrespective of the states of information from its numerical pre-set. After firing, the pre-set will remain intact in terms of information.*



$f^{p_1}(\mathbf{x}_k) \neq 0$ and $f^{t_1}(\mathbf{x}_k) \neq 0$, $\forall \mathbf{x} \in \mathcal{A}_{p_1}, \mathcal{A}_{t_1}$, respectively. Thus: $\int_{\mathcal{B}} \left(f^{p_1} \wedge f^{t_1}\right)(\mathbf{x}_k) d\mathbf{x}_k \neq \emptyset$

(a)



$f^{p_1}(\mathbf{x}_k) \neq 0$ and $f^{t_1}(\mathbf{x}_k) \neq 0$, $\forall \mathbf{x} \in \mathcal{A}_{p_1}, \mathcal{A}_{t_1}$, respectively. $\mathcal{B} = \emptyset$, thus $\int_{\mathcal{B}} \left(f^{p_1} \wedge f^{t_1}\right)(\mathbf{x}_k) d\mathbf{x}_k = 0$

(b)

Fig. 4. Two illustrative examples of a PPNof one transition (t_1) not being fired.

Proof. Let transition t_i be such that $f^{t_i}(\mathbf{x}_k) = \mu(\mathbf{x}_k)$ and $p_j^{(\mathcal{N})}$ a place that belongs to the pre-set of t_i. By definition [20], $\left(f^{p_j} \wedge \mu\right)(\mathbf{x}_k) = f^{p_j}(\mathbf{x}_k) \neq \emptyset$, hence Condition 2 is automatically fulfilled.

Example 1. Definition 7 is conceptually illustrated using two examples in Fig. 4. For simplicity, the net architecture for the PPN of this example is the same as that used in Fig. 3. Note that in panel (a), t_1 is not enabled due to insufficient tokens in $p_1^{(\mathcal{S})}$, whereas in (b) the conjunction $\left(f^{p_1} \wedge f^{t_1}\right)(\mathbf{x}_k)$ is not possible. $\mathcal{A}_{p_1}$ and $\mathcal{A}_{t_1}$ from Fig. 4 represent the subsets of $\mathcal{X}$ where the densities $f^{p_1}(\mathbf{x}_k)$ and $f^{t_1}(\mathbf{x}_k)$ are defined, respectively.

Definition 8 (Information flow dynamics)**.** The firing of any transition t_i always consumes from all its input arcs at the same time. The following are rules which also apply when a numerical or mixed transition $t_i \in \mathbf{T}^{(\mathcal{N})}$ is fired:

1. An input arc from place $p_j^{(\mathcal{N})}$ to transition t_i conveys a state of information given by $a_{ij}^{-}\left(f^{p_j} \wedge f^{t_i}\right)(\mathbf{x}_k)$, which remains in $p_j^{(\mathcal{N})}$ after transition t_i has fired;
2. An output arc from t_i to $p_j^{(\mathcal{N})} \in t_i^{\bullet}$ conveys a state of information given by $a_{ij}^{+}\left(f^{\bullet t_i} \wedge f^{t_i}\right)(\mathbf{x}_k)$, where $f^{\bullet t_i}(\mathbf{x}_k)$ is the resulting density from the disjunction of the states of information of the pre-set of t_i. As stated by Eq. (9), the normalised version of $f^{\bullet t_i}(\mathbf{x}_k)$ can be obtained as:
$$f^{\bullet t_i}(\mathbf{x}_k) = \frac{1}{\beta}\left(f^{p_1} + f^{p_2} + \cdots + f^{p_m}\right)(\mathbf{x}_k) \tag{12}$$
where $p_1, p_2, \ldots, p_m \in {}^{\bullet} t_i$ are numerical places from the pre-set of t_i and β is a normalising constant;
3. The state of information resulting in place $p_j^{(\mathcal{N})} \in t_i^{\bullet}$ after firing t_i, is the disjunction of the states of information $f^{p_j}(\mathbf{x}_k)$, the previous state of information, and $a_{ij}^{+}\left(f^{t_i} \wedge f^{\bullet t_i}\right)(\mathbf{x}_k)$, the information produced after firing transition t_i. Mathematically $f^{p_j}(\mathbf{x}_{k+1}) = \left(f^{p_j} \vee a_{ij}^{+}\left(f^{t_i} \wedge f^{\bullet t_i}\right)\right)(\mathbf{x}_k)$.

Example 2. Fig. 5 provides an example of rules (1)–(3) by using a PPN model of one transition, three numerical places, and two symbolic places. The markings at times k and $k+1$ are highlighted by the panels. At $k+1$, the places $p_1^{(\mathcal{N})}$ and $p_2^{(\mathcal{N})}$ are updated with the information coming from transition t_1 through a conjunction of states of information weighted according to (a_{11}^{-}, a_{12}^{-}), respectively. Observe also that the state of information resulting in place $p_3^{(\mathcal{N})}$ after firing transition t_1, is the joint information between the information that existed in $p_3^{(\mathcal{N})}$ at k, and that produced by transition t_1 through the intersection with its pre-set, i.e. $a_{13}^{+}\left(f^{t_1} \wedge f^{\bullet t_1}\right)(\mathbf{x}_k) = a_{13}^{+}\left(f^{t_1} \wedge (f^{p_1} \vee f^{p_2})\right)(\mathbf{x}_k)$.

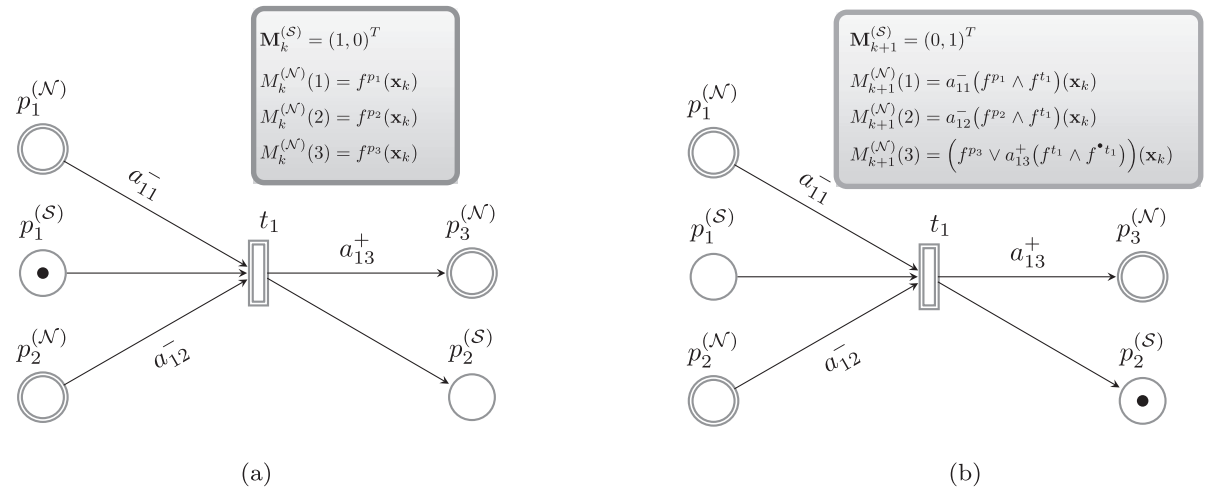


Fig. 5. Illustration of the information flow dynamics of PPNs. (a) Time k, before firing transition t_1. (b) After firing transition t_1 at time $k+1$.

Theorem 1 (Pre-set maximum information content)**.** *Under the assumption that $\mathcal{X}$ is a linear space, then there exists an upper bound value of the information content that will arise at time $k+1$ in the numerical places from the pre-set of transition $t_i \in \mathbf{T}^{(\mathcal{N})}$ after firing at time k. It can be obtained as:*

$$ {}^{\bullet}I_{\max,t_i} = \log\left[\prod_{j=1}^{m}\left(\frac{a_{ij}^-}{\mu^4\alpha_{ij}^3}\mathbb{E}_{f^{t_i}}\left[\left(f^{p_j}\right)^2\right]\mathbb{E}_{f^{p_j}}\left[\left(f^{t_i}\right)^2\right]\right)^{a_{ij}^-}\right] \tag{13} $$

where $\mathbb{E}_{f^{t_i}}[\cdot]$, $\mathbb{E}_{f^{p_j}}[\cdot]$, represent the expectation with respect to f^{t_i} and f^{p_j} respectively, $\mu = cte$ is the homogeneous density function of $\mathcal{X}$, and α_{ij} are the normalising constants resulting from the conjunctions $\left(f^{t_i} \wedge f^{p_j}\right)(\mathbf{x}_k)$. Places $p_1, \ldots, p_j, \ldots, p_m \in \mathbf{P}^{(\mathcal{N})}$ belong to the pre-set of transition t_i.

Proof. See Appendix A.

Remark 1. ${}^{\bullet}I_{\max,t_i}$ represents the maximum learning capacity of the pre-set of transition t_i after being fired at k, which is expressed in terms of amount of information.

Theorem 2 (Post-set maximum information content)**.** *Under the assumption that $\mathcal{X}$ is a linear space, then the maximum information content that will arise in the numerical places from the post-set of $t_i \in \mathbf{T}^{(\mathcal{N})}$ after firing at time k, can be evaluated as:*

$$ I^{\bullet}_{\max,t_i} = \log\left[\prod_{j=1}^{m}\frac{b_{ij}}{\mu}\left(\mathbb{E}_{f^{p_j}}\left[f^{p_j} + a_{ij}^+ f^{\mathrm{conj},t_i}\right]\right)^{b_{ij}}\left(\mathbb{E}_{f^{\mathrm{conj},t_i}}\left[f^{p_j} + a_{ij}^+ f^{\mathrm{conj},t_i}\right]\right)^{1-b_{ij}}\right] \tag{14} $$

where $b_{ij} = \frac{1}{1+a_{ij}^+}$, and $f^{\mathrm{conj},t_i} = \left(f^{t_i} \wedge f^{\bullet t_i}\right)(\mathbf{x}_k)$, the normalised density function resulting from the conjunction of states of information between $f^{t_i}(\mathbf{x}_k)$ and $f^{\bullet t_i}(\mathbf{x}_k)$ (recall Definition 8). Places $p_1, \ldots, p_j, \ldots, p_m \in \mathbf{P}^{(\mathcal{N})}$ belong to the post-set of transition t_i.

Proof. See Appendix B.

Remark 2. $I^{\bullet}_{\max,t_i}$ gives us a measure of the maximum learning capacity of the post-set of transition t_i after being fired at time k.

3.3. PPN dynamics

In PPNs, the marking evolution of the symbolic subnet has been given in Eq. (1). For the dynamics of the numerical subnet, the rules provided in Definition 8 can be applied. Notwithstanding, a more compact definition is possible for the marking evolution of the numerical subnet through a matrix representation, as follows.

Definition 9 (Marking evolution of the numerical subnet)**.** The marking $\mathbf{M}_{k+1}^{(\mathcal{N})}$ can be obtained as a function of the marking at time k through the following state equation:

$$ \mathbf{M}_{k+1}^{(\mathcal{N})} = \left[\mathbf{M}_k^{(\mathcal{N})} \circ \boldsymbol{\gamma}_k + \left(\sum_{i=1}^{n_t^{(\mathcal{N})}}(\mathbf{a}_i^+)^T \otimes \mathbf{c}_i + (\mathbf{A}^-)^T \circ \mathbf{B}\right) \cdot \mathbf{v}_k\right] \circ \boldsymbol{\beta}_k \tag{15} $$

where

1. $\mathbf{v}_k = \left(v_{1,k}, v_{2,k}, \ldots, v_{n_t^{(\mathcal{N})},k}\right)^T$ is the firing vector for the numerical subnet (numerical and mixed transitions) at time k. Its ith element $v_{i,k}$ is equal to 1 if transition t_i is fired, and 0 otherwise;
2. $\mathbf{A}^- = \left[a_{ij}^-\right], i = 1, \ldots, n_t^{(\mathcal{N})},\ j = 1, \ldots, n_p^{(\mathcal{N})}$ is the backward incidence matrix of the numerical subnet;
3. $\mathbf{a}_i^+ = \left(a_{i1}^+, a_{i2}^+, \ldots, a_{in_p^{(\mathcal{N})}}^+\right)^T$, a column vector corresponding to the ith row of the forward incidence matrix of the numerical subnet;
4. $\mathbf{c}_i$ is an $n_t^{(\mathcal{N})}$–dimensional row vector of states of information defined by:[1] $\mathbf{c}_i = (f^{t_i} \wedge f^{\bullet t_i}) \cdot \boldsymbol{\delta}_{i\ell}$, where $\boldsymbol{\delta}_{i\ell}$ is a vector whose elements are the Kronecker delta of variables i and ℓ, which makes all elements zero except for $i = \ell,\ \ell = 1, \ldots, n_t^{(\mathcal{N})}$;
5. $\mathbf{B}$ is an $(n_p^{(\mathcal{N})} \times n_t^{(\mathcal{N})})$ matrix whose (i, j)th element is given by $f^{p_j} \wedge f^{t_i}$;
6. $\boldsymbol{\gamma_k}$ is an $n_p^{(\mathcal{N})}$– dimensional column vector of binary constants, i.e. $\boldsymbol{\gamma_k} = \left(\gamma_k^{(1)}, \ldots, \gamma_k^{(j)}, \ldots, \gamma_k^{(n_p^{(\mathcal{N})})}\right)^T$, where $\gamma_k^{(j)}$ is given by:

$$\gamma_k^{(j)} = \begin{cases} 1, & \text{if} \quad \sum_{i=1}^{n_t^{(\mathcal{N})}} a_{ij}^+ v_{i,k} > 0 \\ 1, & \text{if} \quad \left(\sum_{i=1}^{n_t^{(\mathcal{N})}} a_{ij}^+ v_{i,k} = 0 \ \ \& \sum_{i=1}^{n_t^{(\mathcal{N})}} a_{ij}^- v_{i,k} = 0\right) \\ 0, & \text{if} \quad \left(\sum_{i=1}^{n_t^{(\mathcal{N})}} a_{ij}^+ v_{i,k} = 0 \ \ \& \sum_{i=1}^{n_t^{(\mathcal{N})}} a_{ij}^- v_{i,k} > 0\right); \end{cases} \tag{16}$$

7. $\boldsymbol{\beta_k} = \left(\beta_k^{(1)}, \ldots, \beta_k^{(j)}, \ldots, \beta_k^{(n_p^{(\mathcal{N})})}\right)^T$, a vector of numerical constants defined as:

$$\beta_k^{(j)} = \frac{1}{\gamma_k^{(j)} + \sum_{i=1}^{n_t^{(\mathcal{N})}} (a_{ij}^+ + a_{ij}^-) v_{i,k}} \tag{17}$$

In Eq. (15), the symbols $\langle \otimes, \circ, \cdot \rangle$ are used to denote the outer product, the Hadamard product, and the inner product of matrices [44], respectively. Observe that the summation of outer products $\sum_{i=1}^{n_t^{(\mathcal{N})}} (\mathbf{a}_i^+)^T \otimes \mathbf{c}_i$ in Eq. (15) renders an $(n_p^{(\mathcal{N})} \times n_t^{(\mathcal{N})})$ matrix, i.e.:

$$\sum_{i=1}^{n_t^{(\mathcal{N})}} (\mathbf{a}_i^+)^T \otimes \mathbf{c}_i = \begin{pmatrix} a_{11}^+ f^{t_1} \wedge f^{\bullet t_1} & a_{21}^+ f^{t_2} \wedge f^{\bullet t_2} & \cdots & a_{n_t^{(\mathcal{N})}1}^+ f^{t_{n_t^{(\mathcal{N})}}} \wedge f^{\bullet t_{n_t^{(\mathcal{N})}}} \\ a_{12}^+ f^{t_1} \wedge f^{\bullet t_1} & a_{22}^+ f^{t_2} \wedge f^{\bullet t_2} & \cdots & a_{n_t^{(\mathcal{N})}2}^+ f^{t_{n_t^{(\mathcal{N})}}} \wedge f^{\bullet t_{n_t^{(\mathcal{N})}}} \\ \vdots & & \ddots & \\ a_{1n_p^{(\mathcal{N})}}^+ f^{t_1} \wedge f^{\bullet t_1} & a_{2n_p^{(\mathcal{N})}}^+ f^{t_2} \wedge f^{\bullet t_2} & \cdots & a_{n_t^{(\mathcal{N})} n_p^{(\mathcal{N})}}^+ f^{t_{n_t^{(\mathcal{N})}}} \wedge f^{\bullet t_{n_t^{(\mathcal{N})}}} \end{pmatrix} \tag{18}$$

such that its (i, j)th element is a weighted density function that represents the state of information added to output place $p_j^{(\mathcal{N})}$ after transition t_i has been fired.

Lemma 2. *The density functions within marking $\mathbf{M}_{k+1}^{(\mathcal{N})}$ obtained using Eq. (15) are normalised provided that $\mathbf{M}_k^{(\mathcal{N})}$ is also comprised of normalised density functions.*

Proof. See Appendix C.

Observe that $\boldsymbol{\beta_k}$ acts as a vector of normalising constants required for $\mathbf{M}_{k+1}$ to be a vector of bona fide densities. Observe also that the evaluation of the referred normalising constants can be bypassed when using a particle method for the approximation of the density functions, as explained in Section 2.

Example 3. Fig. 6 represents a PPN which is used here to illustrate the PPN dynamics described above. The initial marking $\mathbf{M}_0^{(\mathcal{S})}$ comprises three tokens in $p_1^{(\mathcal{S})}$. The probability densities for marking $\mathbf{M}_0^{(\mathcal{N})}$ are Gaussians given by: $f_0^{p_1} = \mathcal{N}(20, 5)$, $f_0^{p_2} = \mathcal{N}(15, 5)$, $f_0^{p_3} = \mathcal{N}(10, 5)$. Transition t_1 is a mixed transition such that $f^{t_1} = \mathcal{N}(25, 5)$. According to the graph given by Fig. 6, the incidence matrices used for calculations of the numerical and symbolic subnets are:

$$\mathbf{A}^{(\mathcal{N})} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 2 & 0 \end{pmatrix} = \begin{pmatrix} -1 & -2 & 1 \end{pmatrix} \tag{19a}$$

[1] Hereinafter, the argument $(\mathbf{x}_k)$ from $f(\mathbf{x}_k)$ is dropped for clarity.

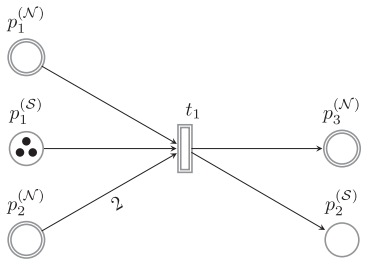


Fig. 6. PPN of the Example 3.

$$\mathbf{A}^{(\mathcal{S})} = \begin{pmatrix} 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 1 \end{pmatrix} \tag{19b}$$

respectively. In this example, matrix $\mathbf{B}$ is calculated as:

$$\mathbf{B} = \begin{pmatrix} f^{p_1} \wedge f^{t_1} \\ f^{p_2} \wedge f^{t_1} \\ f^{p_3} \wedge f^{t_1} \end{pmatrix} \tag{20}$$

and hence

$$(\mathbf{A}^-)^T \circ \mathbf{B} = \begin{pmatrix} 1 \cdot f^{p_1} \wedge f^{t_1} \\ 2 \cdot f^{p_2} \wedge f^{t_1} \\ 0 \cdot f^{p_3} \wedge f^{t_1} \end{pmatrix} \tag{21}$$

On the other hand, vector $\mathbf{c}_i$ here is a one-dimensional row vector given by $(f^{t_1} \wedge f^{\bullet t_1}) = (\mathcal{N}(25,5) \wedge f^{\bullet t_1})$, where

$$f_0^{\bullet t_1} = f^{p_1} \vee f^{p_2} = 1/2\big(\mathcal{N}(15,5) + \mathcal{N}(20,5)\big) \tag{22}$$

Next, the term $\sum_{i=1}^{n_t^{(\mathcal{N})}} (\mathbf{a}_i^+)^T \otimes \mathbf{c}_i$ from Eq. (15) can be obtained as:

$$\sum_{i=1}^{n_t^{(\mathcal{N})}} (\mathbf{a}_i^+)^T \otimes \mathbf{c}_i = (\mathbf{a}_1^+)^T \otimes \mathbf{c}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \left(f^{t_1} \wedge f_0^{\bullet t_1} \right) = \begin{pmatrix} 0 \cdot f^{t_1} \wedge f_0^{\bullet t_1} \\ 0 \cdot f^{t_1} \wedge f_0^{\bullet t_1} \\ 1 \cdot f^{t_1} \wedge f_0^{\bullet t_1} \end{pmatrix} \tag{23}$$

As a first step, we want to evaluate the marking at $k = 1$. Eqs. (1) and (15) are applied in confluence with the rule for transition firing (recall Definition [7]) for the system state evolution $(k = 0 \to k = 1)$ as follows:

$$\mathbf{M}_1^{(\mathcal{N})} = \left(\begin{pmatrix} \gamma_0^{(1)} \cdot f_0^{p_1} \\ \gamma_0^{(2)} \cdot f_0^{p_2} \\ \gamma_0^{(3)} \cdot f_0^{p_3} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ f^{t_1} \wedge f_0^{\bullet t_1} \end{pmatrix} + \begin{pmatrix} f_0^{p_1} \wedge f^{t_1} \\ 2 \cdot f_0^{p_2} \wedge f^{t_1} \\ 0 \end{pmatrix} \right) \circ \begin{pmatrix} \beta_0^{(1)} \\ \beta_0^{(2)} \\ \beta_0^{(3)} \end{pmatrix} = \begin{pmatrix} f_0^{p_1} \wedge f^{t_1} \\ \frac{1}{2} \cdot 2 \cdot f_0^{p_2} \wedge f^{t_1} \\ \frac{1}{2} \cdot \left(\left(f^{t_1} \wedge f_0^{\bullet t_1} \right) + f_0^{p_3} \right) \end{pmatrix} \tag{24}$$

where $\boldsymbol{\gamma}_0 = \big(0,0,1\big)^T$ and $\boldsymbol{\beta}_0 = \big(1,1/2,1/2\big)^T$, which have been obtained according to Eqs. (16) and (17), respectively. Thus, by substituting $f_0^{p_j}, j = \{1,2,3\}$ into Eq. (24), the marking $\mathbf{M}_1 = \left(\mathbf{M}_1^{(\mathcal{N})}, \mathbf{M}_1^{(\mathcal{S})}\right)$ is obtained as follows:

$$\mathbf{M}_1^{(\mathcal{N})} = \begin{pmatrix} \mathcal{N}(20,5) \wedge \mathcal{N}(25,5) \\ \mathcal{N}(15,5) \wedge \mathcal{N}(25,5) \\ \frac{1}{2} \cdot \left(\mathcal{N}(25,5) \wedge f_0^{\bullet t_1} + \mathcal{N}(10,5) \right) \end{pmatrix}, \quad \mathbf{M}_1^{(\mathcal{S})} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \tag{25}$$

where $f_0^{\bullet t_1}$ has been previously defined in Eq. (22). Next, the process is repeated until the PPN stops at $k = 3$, however the analytical expressions have been omitted here for clarity. Instead, the results for the numerical places have been represented for states $k = 0$ to $k = 3$ in Fig. 7. A summary of the results for the analysis of the PPN model, along with some terms used to evaluate Eqs. (1) and (15), are provided in Table 1. Observe that the results for symbolic marking, active transitions, and $\boldsymbol{\gamma}_k$-constants are column vectors although they are not explicitly reflected as so in Table 1 for clarity.

Beside, an exercise is carried out to evaluate the response of the PPN shown in Fig. 6 at $k = 1$ by adopting the homogeneous density function as state of information for t_1, i.e. $f^{t_1} = \mu(x)$. Note that by Lemma 1, transition t_1 is always enabled in this case provided that at least one token is in $p_1^{(\mathcal{S})}$, which occurs for $k = 0$ to $k = 3$. Also, matrix $\mathbf{B} = (f^{p_1}, f^{p_2}, f^{p_3})^T$ and

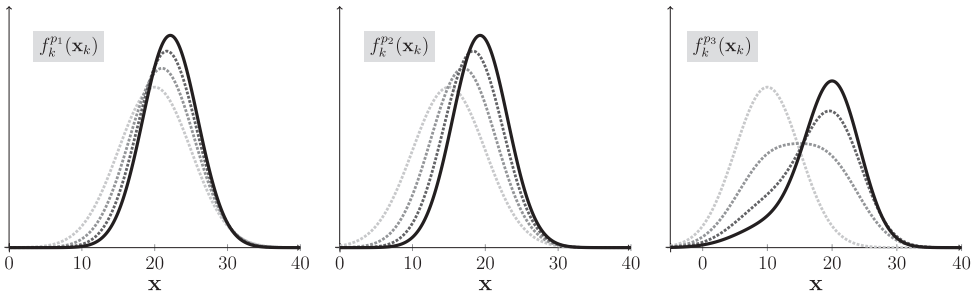


Fig. 7. Results from the evaluation of Eq. (15) over the numerical subnet of the PPN given in Fig. 6. Four states are represented using increasing grey tones until the final state at $k = 3$ (solid line).

Table 1
Some outputs from the evaluation of the PPN shown in Fig. 6.

State	$\mathbf{M}_k^{(\mathcal{S})}$	$\boldsymbol{\gamma}_k$	$\boldsymbol{\beta}_k$	$\mathbf{v}_k$
$k = 0$ (starts)	(3, 0)	(0, 0, 1)	(1, 1/2, 1/2)	(1)
$k = 1$	(2, 1)	(0, 0, 1)	(1, 1/2, 1/2)	(1)
$k = 2$	(1, 2)	(0, 0, 1)	(1, 1/2, 1/2)	(1)
$k = 3$ (stops)	(0, 3)	(0, 0, 1)	(1, 1/2, 1/2)	(0)

$\mathbf{c}_1 = \left(f_0^{\bullet t_1}\right) = (f^{p_1} \vee f^{p_2}) = 1/2(f^{p_1} + f^{p_2})$, hence Eq. (24) can be rewritten as:

$$\mathbf{M}_1^{(\mathcal{N})} = \left(\begin{pmatrix}0\\0\\f_0^{p_3}\end{pmatrix} + \begin{pmatrix}0\\0\\f_0^{\bullet t_1}\end{pmatrix} + \begin{pmatrix}f_0^{p_1}\\2\cdot f_0^{p_2}\\0\end{pmatrix}\right) \circ \begin{pmatrix}1\\\frac{1}{2}\\\frac{1}{2}\end{pmatrix} = \begin{pmatrix}f_0^{p_1}\\f_0^{p_2}\\[1\,pt]\frac{1}{2}\cdot\left(f_0^{\bullet t_1} + f_0^{p_3}\right)\end{pmatrix} \tag{26}$$

which finally leads to:

$$\mathbf{M}_1^{(\mathcal{N})} = \begin{pmatrix}\mathcal{N}(20,5)\\\mathcal{N}(15,5)\\\frac{1}{2}\cdot\left(f_0^{\bullet t_1} + \mathcal{N}(10,5)\right)\end{pmatrix},\quad \mathbf{M}_1^{(\mathcal{S})} = \begin{pmatrix}2\\1\end{pmatrix} \tag{27}$$

where $f_0^{\bullet t_1} = 1/2\big(\mathcal{N}(20,5) + \mathcal{N}(15,5)\big)$. Observe that, according to Lemma 1, the pre-set places $p_1^{(\mathcal{N})}$ and $p_2^{(\mathcal{N})}$ remain intact in terms of information, i.e. their respective states of information are invariant after firing t_1.

4. PPN algorithm

The analysis of the numerical part of a PPN requires the evaluation of conjunction of states of information which are affected by normalising constants (recall Eq. (5)). The calculation of such normalising constants involves the evaluation of integrals which are intractable except for exceptionally simple nets using some parametric density functions as states of information (e.g. Gaussians). To alleviate this drawback and confer the required versatility to our PPN methodology, particle methods are proposed to approximate the conjunction and disjunction of states of information, as described in Section 2.2. In this section, a pseudocode implementation of PPNs is provided as Algorithm 3, which combines the PPN methodology with the particle approximation for the conjunction and disjunction of states of information. Three main blocks comprise the pseudocode, namely: *transition firing, information exchange*, and *marking evolution*, which have been highlighted for better clarity. Note that in the PPN algorithm, the normalising constants from Eq. (17) have been omitted since the particle approximation bypasses them through resampling. Otherwise, use Eq. (17) to obtain the corresponding constant after step 32 and modify the output resulting from step 33 accordingly.

5. Application example

The PPN methodology proposed above is exemplified here to illustrate its potential in modelling dynamic hybrid systems. To this end, the ageing of an engineering component or sub-system is proposed to be analysed by an idealised expert system using condition-based monitoring from sensors along with information coming from expert knowledge. Fig. 8 illustrates the idealised expert system through a PPN consisting of five numerical places ($p_1^{(\mathcal{N})}$ to $p_5^{(\mathcal{N})}$), five symbolic places ($p_1^{(\mathcal{S})}$ to $p_5^{(\mathcal{S})}$), four mixed transitions (t_1 to t_4), and one symbolic transition labelled as t_5. The numerical places comprise

Algorithm 3 Plausible Petri nets algorithm.

Inputs: $\mathfrak{M} = \left\langle \mathbf{P}, \mathbf{T}, \mathbf{F}, \mathbf{W}, \mathbf{D}, \mathcal{X}, \mathcal{G}, \mathcal{H}, \mathbf{M}_0 \right\rangle$, ▷ {Tuple of defining elements for the PPN}
Outputs: $\mathbf{M}_k$

begin ($k \geqslant 0$):

Set $\tilde{f}_k^{p_j} = (\emptyset)$, $j = 1, \ldots, n_p^{(\mathcal{N})}$ ▷ {Auxiliary densities}
▷ Transition firing (recall Def. 7)
for all $t_i \in \mathbf{T}$ **do**
  Require $\tau_i \in \mathbf{D}$ has passed
  Initialise $u_{i,k}$, $v_{i,k}$ as $u_{i,k} = 1,\ v_{i,k} = 1$
  for all $p_j \in {}^\bullet t_i$ **do**
    switch t_i **do**
      case $t_i \in \mathbf{T}^{(\mathcal{S})}$
        if $M_k^{(\mathcal{S})}(j) < a'_{ij}$ **then** $u_{i,k} = 0$
        end if
      case $t_i \in \mathbf{T}^{(\mathcal{N})}$
        if $f^{p_j} \wedge f^{t_i} = \emptyset$ **then** $v_{i,k} = 0$
        end if
      case $t_i \in \mathbf{T}^{(\mathcal{N})} \cap \mathbf{T}^{(\mathcal{S})}$
        if $u_{i,k} = 0$ **or** $v_{i,k} = 0$ **then** $u_{i,k} = 0,\ v_{i,k} = 0$
        end if
  end for
  ▷ Information exchange
  if $t_i \in \mathbf{T}^{(\mathcal{N})}$ **then**
    Obtain $f^{{}^\bullet t_i}$ ▷ {Use Algorithm 2}
    Evaluate $f^{{}^\bullet t_i} \wedge f^{t_i}$ ▷ {Use Algorithm 2}
    for all $p_j^{(\mathcal{N})} \in \mathbf{P}^{(\mathcal{N})}$ **do**
      Evaluate $f^{t_i} \wedge f^{p_j}$ ▷ {Use Algorithm 1}
      Set $\tilde{f}_k^{ij} \leftarrow a_{ij}^- \left(f^{t_i} \wedge f^{p_j} \right),\ a_{ij}^- \in \mathbf{W}^{(\mathcal{N})}$
      Set $\tilde{f}_k^{ii} \leftarrow a_{ij}^+ \left(f^{{}^\bullet t_i} \wedge f^{t_i} \right),\ a_{ij}^+ \in \mathbf{W}^{(\mathcal{N})}$
      Update $\tilde{f}_k^{p_j} \leftarrow \tilde{f}_k^{p_j} \vee v_{i,k} \left(\tilde{f}_k^{ij} \vee \tilde{f}_k^{ii} \right)$
    end for
  end if
end for
▷ Marking evolution
for all $p_j^{(\mathcal{N})} \in \mathbf{P}^{(\mathcal{N})}$ **do**
  Evaluate $\gamma_k^{(j)}$ from Eq. (16)
  Obtain $f_{k+1}^{p_j} \leftarrow \gamma_k^{(j)} f_k^{p_j} \vee \tilde{f}_k^{p_j}$
  Set $M_{k+1}^{(\mathcal{N})}(j) \leftarrow f_{k+1}^{p_j}$
end for
for all $p_j^{(\mathcal{S})} \in \mathbf{P}^{(\mathcal{S})}$ **do**
  $M_{k+1}^{(\mathcal{S})}(j) \leftarrow M_k^{(\mathcal{S})}(j) + \sum_{i=1}^{n_t^{(\mathcal{S})}} a'_{ji} u_{i,k},\ a'_{ji} \in \mathbf{W}^{(\mathcal{S})}$
end for
Set $\mathbf{M}_{k+1}^{(\mathcal{S})} = \left(M_{k+1}^{(\mathcal{S})}(1), \ldots, M_{k+1}^{(\mathcal{S})}(n_p^{(\mathcal{S})}) \right)^T$
Set $\mathbf{M}_{k+1}^{(\mathcal{N})} = \left(M_{k+1}^{(\mathcal{N})}(1), \ldots, M_{k+1}^{(\mathcal{N})}(n_p^{(\mathcal{N})}) \right)^T$
$\mathbf{M}_{k+1} = \left(\mathbf{M}_{k+1}^{(\mathcal{N})}, \mathbf{M}_{k+1}^{(\mathcal{S})} \right)$
$\mathbf{M}_k \leftarrow \mathbf{M}_{k+1}$

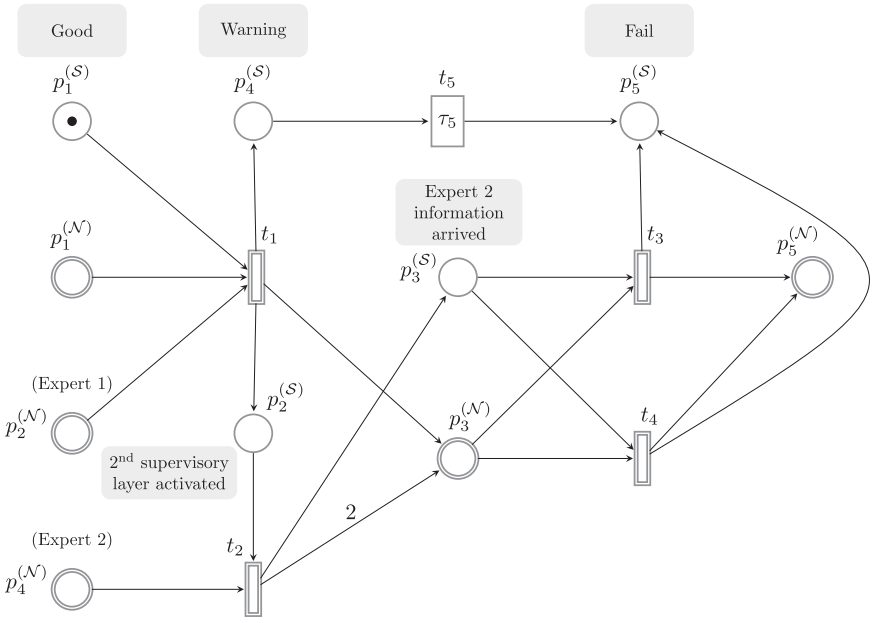


Fig. 8. Plausible Petri net of the example in Section 5.

Table 2
Description of the numerical places of the PPN shown in Fig. 8.

ID	Source of information	Initial state of information	Dynamic equation
$p_1^{(\mathcal{N})}$	Sensor $\mathtt{s}$	$f_0^{p_1} \sim \mathcal{N}(20,\ 3)$	Eq. (28)
$p_2^{(\mathcal{N})}$	Expert 1	$f_0^{p_2} \sim \mathcal{N}(16,\ 2)$	–
$p_3^{(\mathcal{N})}$	–	$(\emptyset)$	Eq. (28)
$p_4^{(\mathcal{N})}$	Expert 2	$f_0^{p_4} \sim \mathcal{N}(16,\ 1)$	–
$p_5^{(\mathcal{N})}$	–	$(\emptyset)$	–

the states of information about a state variable $x_k \in \mathbb{R}$, given by the PDFs $f_k^{p_1}$ to $f_k^{p_5}$ respectively. In this example, the state variable in places $p_1^{(\mathcal{N})}$ and $p_3^{(\mathcal{N})}$ is assumed to evolve over time $k \in \mathbb{N}$ following a dynamic equation $x_k = h_k(x_{k-1}, \theta)$ given by:

$$x_k = e^{-\theta k} x_{k-1} + w_k \tag{28}$$

where θ is an uncertain decay parameter whose values are modelled as a Gaussian centred at 0.005 with 50% of coefficient of variation. The term w_k represents white-noise type measurement error from a sensor $\mathtt{s}$, with standard deviation given by $\sigma_{\mathtt{s}} = 1$. In view of Fig. 8, the proposed expert system also encompasses information from experts in places $p_2^{(\mathcal{N})}$ and $p_4^{(\mathcal{N})}$, represented by the PDFs $f_k^{p_2}$ and $f_k^{p_4}$, respectively. Table 2 lists the initial state of information of x_k within the numerical places, along with the corresponding function representing the dynamics of x_k for each place. With the proposed system configuration, nodes $\{p_1^{(\mathcal{S})}, p_1^{(\mathcal{N})}, p_2^{(\mathcal{N})}, t_1\}$ are intended to constitute the first supervisory layer for the state variable x_k.

In this example, the mixed transitions t_i, $i = \{1, 3, 4\}$, are defined based on condition (recall Definition 5) using indicator functions for the regions of the state space $\mathcal{C}_i \subseteq \mathcal{X}$, i.e., $f^{t_i} \sim \mathbb{I}_{\mathcal{C}_i}(x_k)$, that assigns a value of 1 when $x_k \in \mathcal{C}_i$ and 0 otherwise, where:

$$\mathcal{C}_1 = \left\{ x_k \in \mathcal{X} : \mathbb{E}_{f^{p_1}}(x_k) \leqslant \epsilon_1 \right\} \tag{29a}$$

$$\mathcal{C}_3 = \left\{ x_k \in \mathcal{X} : \mathbb{E}_{f^{p_3}}(x_k) \leqslant \epsilon_2 \right\} \tag{29b}$$

$$\mathcal{C}_4 = \left\{ x_k \in \mathcal{X} : H(x_k) > \xi \right\} \tag{29c}$$

In Eqs. (29a) and (29b), $\mathbb{E}_{f^{p_j}}(x_k)$ denotes the expectation of x_k with respect to f^{p_j}, $j = \{1, 3\}$ respectively, where $\epsilon_1 = 17$, $\epsilon_2 = 7$, and $\xi = 2.25$, are threshold values. The function $H : \mathcal{X} \to \mathbb{R}$ in Eq. (29c) denotes the differential entropy (DE) of

Table 3
Description of the transitions of the PPN shown in Fig. 8.

ID	Type	Rule	State of information	Action
t_1	Mixed	$\mathbb{E}_{f^{p_1}}(x_k) \leqslant 17$	$f^{t_1} \sim \mathbb{I}_{\mathcal{C}_1}(x_k)$	Switches to "Warning condition"
t_2	Numerical	—	$f^{t_2} \sim \mu(x_k)$	Aggregates Expert 2 information
t_3	Mixed	$\mathbb{E}_{f^{p_3}}(x_k) \leqslant 7$	$f^{t_3} \sim \mathbb{I}_{\mathcal{C}_3}(x_k)$	Switches to "Fail condition"
t_4	Mixed	$H(x_k) > 2.25$	$f^{t_4} \sim \mathbb{I}_{\mathcal{C}_4}(x_k)$	Switches to "Fail condition"
t_5	Symbolic	$\tau_5 \geq 20$ (delay)	Not applicable	Switches to "Fail condition"

x_k, which is obtained by calculating[2] $1/2\ln[(2\pi e)var(x_k)]$ as a measure quantifying the uncertainty of x_k. Transition t_2 is defined through a homogeneous density function (recall Lemma 1), hence its firing is conditioned upon place $p_2^{(\mathcal{S})}$ receiving a token. An overview of the complete set of transitions is provided in Table 3.

Initially at $k = 0$, the system starts in the "Good" condition represented by one token at $p_1^{(\mathcal{S})}$, thus $\mathbf{M}_0^{(\mathcal{S})} = (1, 0, 0, 0, 0)^T$. Subsequently, the state variable x_k starts evolving over time following the dynamic model given by Eq. (28) in $p_1^{(\mathcal{N})}$. Once the expected value of x_k in place $p_1^{(\mathcal{N})}$ has reached the threshold value ϵ_1, then transition t_1 is fired provided that the conjunction between the information from Expert 1 and t_1 is possible; next, the system turns to a "Warning" condition. At this time, the resulting information from $p_1^{(\mathcal{N})}$ and $p_2^{(\mathcal{N})}$ along with the information from Expert 2, is transferred to place $p_3^{(\mathcal{N})}$. The firing of t_1 activates a second supervisory layer based on nodes $\{p_3^{(\mathcal{N})}, p_4^{(\mathcal{N})}, p_5^{(\mathcal{N})}, t_2, t_3, t_4\}$ along with their corresponding symbolic nodes, as indicated by the graph in Fig. 8. In this second supervisory layer, a decision is made conditioned upon: (1) the quality of the information in $p_3^{(\mathcal{N})}$, 2) the position of the mean of x_k with respect to the threshold value $\epsilon_2 \in \mathbb{R}$, and finally (3) the total time spent by the system under the warning state. The DE, which was defined above, is used as a quality indicator of the information in place $p_3^{(\mathcal{N})}$, so that t_3 is activated if the DE of $x_k \sim f^{p_3}$ is higher than ξ. Note that the degradation process, for which the most up-to-date information after t_1 is fired is given by $f_k^{p_3}$ at $p_3^{(\mathcal{N})}$, will continue until any of the transitions t_3, t_4, t_5 are activated whereupon the system turns to the "Fail" state, and all the information is collected in $p_5^{(\mathcal{N})}$. Finally note also that $p_3^{(\mathcal{S})}$ helps in synchronising transitions t_3 and t_4, so as to avoid them being activated before the information from Expert 2 has arrived in $p_3^{(\mathcal{N})}$.

Algorithm 3 has been applied to evaluate the system state evolution described through the marking $\mathbf{M}_k, k > 0$. A total of $N = 1000$ particles have been used to approximate the conjunction and disjunction of states of information using Algorithms 1 and 2, respectively. The results for the numerical places $p_1^{(\mathcal{N})}$ and $p_3^{(\mathcal{N})}$ are depicted in Fig. 9 for $k = 0$ to $k = 25$ (see panels from the left). Observe that at $k = 9$, transition t_1 is fired since the conjunction $\left(f^{p_1} \wedge \mathbb{I}_{\mathcal{C}_1}\right)(x_k) = f^{p_1}\left(x_k | \mathbb{E}(x_k) \leqslant 17\right) \neq \emptyset$ and the other transition firing conditions are met (recall Definition 7). Once t_1 is fired, then the information coming from Expert 2 is incorporated and the resulting state of information is subsequently transferred to $p_3^{(\mathcal{N})}$ (see Fig. 9c). According to the proposed PPN methodology and considering the net architecture in Fig. 8, the state of information arising at $p_3^{(\mathcal{N})}$ ($k = 10$) is equal to $\left((f^{p_1} \vee f^{p_2}) \wedge \mathbb{I}_{\mathcal{C}_1}\right) \vee 2f^{p_4} = 1/4\left(f\left(x_k | \mathbb{E}(x_k) \leqslant 17\right) + f^{p_2}(x_k) + 2f^{p_4}(x_k)\right)$. Observe also from Fig. 9c the higher plausibility values shown through darker grey tones, which reveal the influence of the information from Expert 2 on reducing the uncertainty. Note finally from Fig. 9c that transition t_3 is activated at $k = 15$ because the condition $x_k \in \mathcal{C}_4$ is met at that time, i.e. the DE of the state of information given by f^{p_3} is higher than $\xi = 2.25$ at $k = 15$, as shown in Fig. 9e. Next, the system turns to the "Fail" condition and the joint state of information about x_k is finally collected in place $p_5^{(\mathcal{N})}$. A summary of the results for the analysis of the PPN model is provided in Table 4. The symbolic marking and active transitions (2nd and 3rd column of Table 4) are column vectors, although they are not explicitly reflected as so in this table for clarity. The fourth column indicates the sequence of main events, like activation of the information from experts and/or firing of numerical transitions. A representative set of CPU times required by Algorithm 3 to evaluate the kth step of the PPN model shown in Fig. 8, are indicated in the sixth column of Table 4. These values, which have been obtained using a 3.5 GHz double-core system, reveal that the PPN algorithm can efficiently evaluate the marking evolution of a PPN as the one shown in Fig. 8 with a feasible computational burden.

6. Discussion

In Section 5, an application example has been provided to illustrate the dynamics of the proposed PPNs and the different types of information that can be managed, for example, uncertain information from sensors and expert opinions, which confers to PPNs the versatility to solve problems by reasoning about uncertain knowledge. In this section, a comparative analysis is carried out by examining the behaviour of the PPN depicted in Fig. 8 under different scenarios, in order to confer a basis for discussion. As a first exercise, the influence of the uncertainty about x_k on the system response is investigated. To this end, the PPN is evaluated by considering the same configuration as in Section 5 with the exception that the information

[2] This expression for the differential entropy is actually an upper-bound approximation to the actual differential entropy, where the exactness is achieved when the density function is Gaussian.

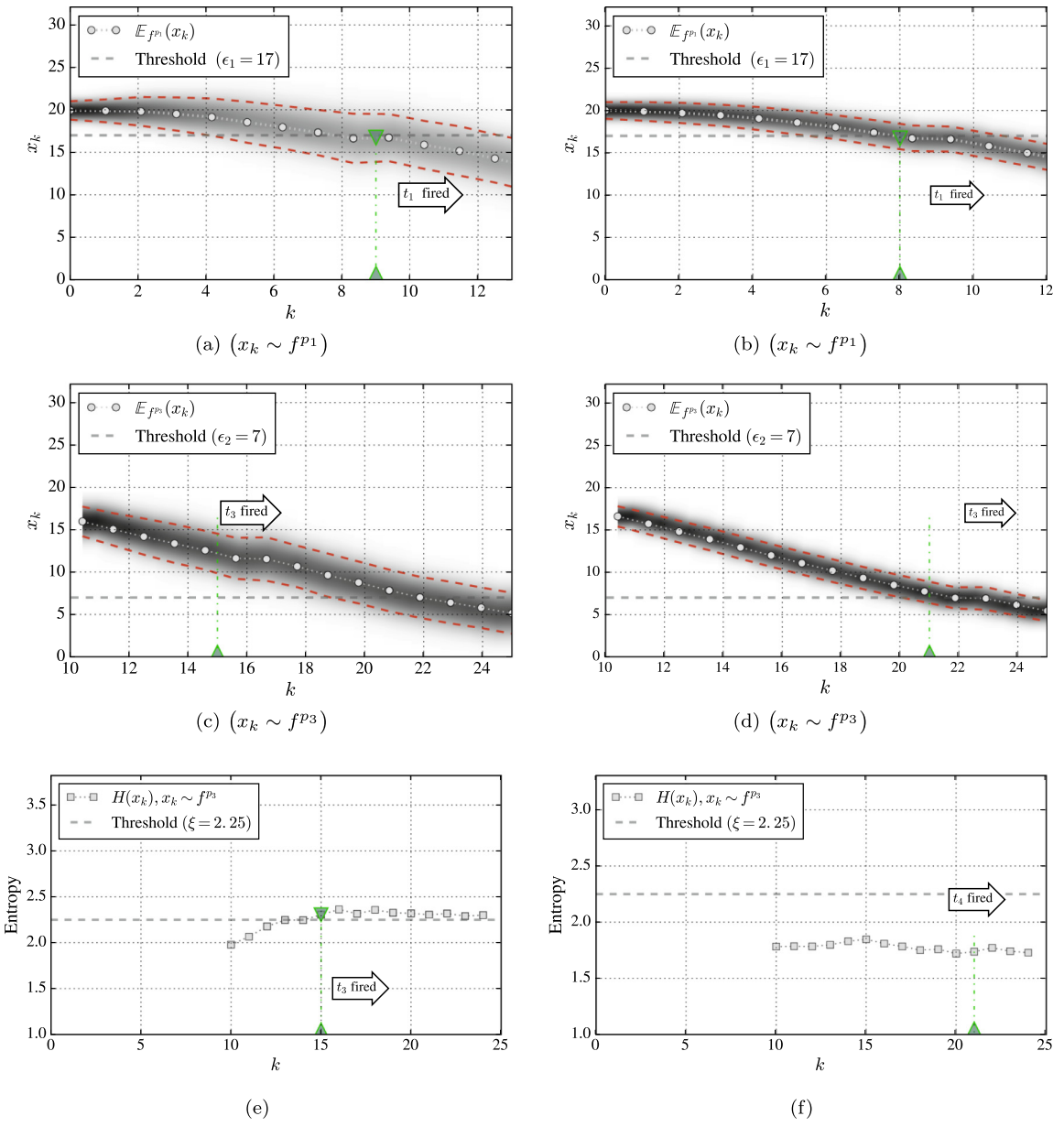


Fig. 9. Representation of the evolution over time of the states of information about the variable x_k in places $p_1^{(\mathcal{N})}$ (panels [a] and [b]), and $p_3^{(\mathcal{N})}$ (panels [c] and [d]). The uncertainty evolution is represented using grey shadows and red dash-dot lines over the $\pm$ std probability band. In lower panels ([e] and [f]), the differential entropy (DE) of $x_k \sim f^{p_3}$ is shown using a dotted-squared line. The green triangles are used to help visualise the time when a transition is activated based on the condition being monitored, as described in the text. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

from sensor $\mathtt{s}$ comes with a white-noise type error $\sigma_{\mathtt{s}} = 0.5$, i.e., half of the error initially considered in Section 5. The results are shown in Fig. 9 (right-side panels). Note from panels (b) and (d) that darker grey tones are obtained as a result of the lower dispersion of the PDFs occurring at respective places $p_1^{(\mathcal{N})}$ and $p_3^{(\mathcal{N})}$, because of the reduced noise. The latter can be corroborated by comparatively examining the DE plots from panels (e) and (f). In terms of PPN behaviour, note also that the transition t_4 is fired since condition C_3 is met, which means that, in this case, the system identifies the Fail state because the expected information from $p_3^{(\mathcal{N})}$ has reached the second threshold $\epsilon_2 = 7$ within the period of evaluation $k = 0 \rightarrow 25$, whilst the DE remains lower than the specified threshold $\xi = 2.25$ for this period, as shown in Fig. 9d and 9 f. Observe that the response of the PPN to such a change in the system configuration (which might represent in practice the renewal of sensor $\mathtt{s}$ to an upgraded version) is different from the one shown in Section 5, where the Fail state was identified due to the high uncertainty about x_k, then transition t_3 was fired. These results are satisfactory in the sense that they confirm that

Table 4
Summary of the results from the analysis of the PPN shown in Fig. 8 using Algorithm 3.

State	$\mathbf{M}_k^{(\mathcal{S})}$	$\mathbf{v}_k$	Event	State	CPU time [s]
$k=0$	(10000)	(00000)	–	Good	0.79
$k=1$	(10000)	(00000)	–	Good	0.82
⋮		⋱		⋮	
$k=9$	(10000)	(10000)	t_1 fired	Good	0.91
$k=10$	(01010)	(01000)	Activate E_2	Warning	0.79
$k=11$	(00110)	(00000)	–	Warning	0.78
⋮		⋱		⋮	
$k=15$	(00110)	(00100)	t_3 fired	Warning	0.81
$k=16$	(00001)	(00000)	–	Fail	0.82

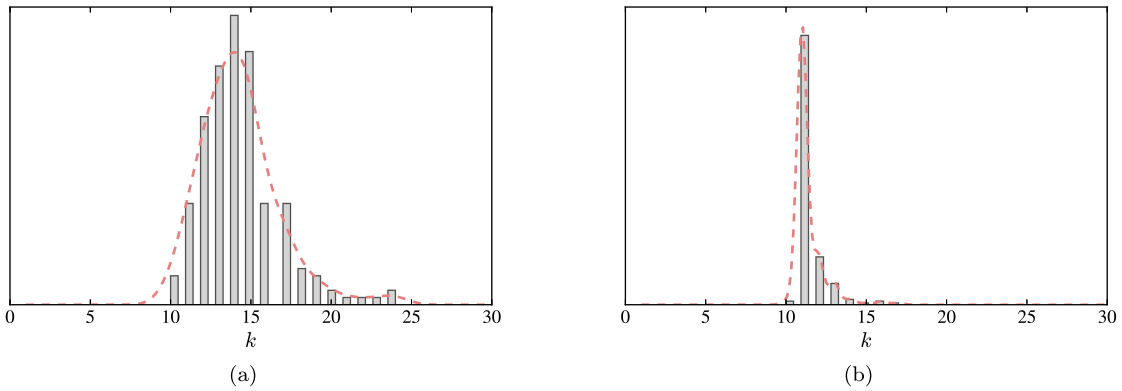


Fig. 10. Panel (a): Histogram representation of the distribution of discrete times when the PPN from Fig. 8 reaches the Fail state. In panel (b), the results are obtained for the case that expert information from $p_4^{(\mathcal{N})}$ is null, i.e. $f^{p_4} = \emptyset$. Kernel density estimates are shown using dashed-red lines. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

a PPN can autonomously adapt its behaviour based not only on the numerical values of the state variable, but also on its information content (e.g. the amount of uncertainty).

Next, as a second discussion exercise, a numerical experiment is carried out to investigate the influence of expert knowledge in the system response. In particular, the PPN shown in Fig. 8 is comparatively evaluated with and without consideration of the information from Expert 2 by choosing $f^{p_4} = \mathcal{N}(16, 1)$ and $f^{p_4} = \emptyset$, respectively. The interest here is in comparatively obtaining the time when the expert system identifies the Fail state under both situations. Fig. 10 shows the results obtained considering 200 independent runs of the PPN algorithm. The results reveal that the information from Expert 2 delays the PPN in reaching the Fail state with respect to the case when information from Expert 2 is absent. Such behaviour can be explained in terms of the uncertainty present in $p_3^{(\mathcal{N})}$ with and without information from Expert 2, such that when $f^{p_4} = \emptyset$, then the threshold $\xi = 2.25$ is reached prematurely in comparison to the case when $f^{p_4} = \mathcal{N}(16, 1)$. This result corroborates the role of the information from Expert 2 on reducing the uncertainty, as explained in the last section. Both exercises, together with the results given in Section 5, reveal the influence of the information and its uncertainty in the overall response of the system represented by the PPN, and in particular reveal how the numerical and symbolic subnets interact to control the execution of the overall system.

In addition to the numerical exercises explained above, a comparative study is considered here to enrich the discussion and to highlight the main differences between our PPNs and the most relevant PN paradigms from the literature. The comparison is carried out in terms of the knowledge represented, uncertainty, type of tokens, available matrix representation of the marking evolution, and possibility to represent time. The results are summarised in Table 5. Note that one of the main differences to remark from Table 5 is that, unlike classical approaches like PNs, SPNs, and HPNs, PPNs can handle uncertainty within their formulation. However, in the author's opinion, the most important advantage in this context is the way that PPNs represent the uncertainty through *states of information* acting as moving tokens for the numerical subnet. This confers the flexibility of our PPNs to represent uncertain knowledge in a more principled approach (i.e., the information from a sensor or from an expert is more easily and versatilely represented through a PDF, rather than through a fuzzy set, like in FPNs, or through a possibility distribution, like in π-PNs). Also, the use of PDFs as tokens enables us to use the principles of information theory and statistics to treat, process, analyse, and combine these PDFs for our convenience, for example, when modelling the information flow between nodes. Moreover, unlike the HPN paradigm, PPNs allow the numerical and symbolic

Table 5
Synoptic table about characteristics of main Petri nets variants.

Paradigm	Kind of information represented	Uncertainty	Tokens	Matrix representation	Time factor
PNs	Discrete	Not considered	Dots, as moving units	Yes	Considered in Timed Petri nets
SPNs	Discrete	Partially considered through probability of firing of transitions using PDFs	Dots, as moving units	Yes	Considered in Timed Stochastic Petri nets
FPNs	Fuzzy production rules. Imprecise or vague information about truth of propositions	Fuzzy theory	Dots (only one token is allowed per place). Tokens are associated to truth of propositions	Not generally used, however several works provide matrix representation, e.g. [12,45–47]	Not generally considered except for Timed Fuzzy Petri nets [48]
π-PNs	Distributions indicating possibility of an imprecise proposition	Based on Dubois and Prade's possibility theory [49]	Dots as possibilistic tokens. Tokens distributed over a set of possible locations with associated degree of truth	No	Not considered
HPNs	Combination of discrete and continuous processes (hybrid systems)	Not generally considered, except in particular versions like in [30,31]	Real values (numerical subnet) and dots (symbolic subnet)	No	Considered in Timed Hybrid Petri nets [7]
HPPNs	Combination of discrete and continuous processes (hybrid systems), approximated by a limited set of particles	Based on weighted particles for numerical values. Symbolic uncertainty is also considered by pseudo-firing of tokens	Can be numerical (particles), symbolic, and hybrid tokens	No	Naturally considered through the dynamic equations of the numerical subnet
PPNs	Combination of discrete and uncertain continuous processes (hybrid systems)	Degree of belief of the possible (plausible) values of uncertain variables that represent the continuous processes [39,42]	States of information [20] (PDFs) for numerical subnet, and dots (for the symbolic subnet)	Yes	Naturally considered through the dynamic equations of the numerical subnet

PNs: Classical Petri nets, SPNs: Stochastic Petri nets, FPNs: Fuzzy Petri nets, π-PNs: Possibilistic Petri nets, HPNs: Hybrid Petri nets, HPPNs: Hybrid Particle Petri nets.

subnets to interact in a bidirectional manner, as was previously mentioned above and also in Section 1, which confers the modeller higher versatility to represent hybrid engineering systems.

7. Conclusions

A novel hybrid approach for PNs has been proposed in this paper, which has been named Plausible Petri nets (PPNs), since it can effectively represent plausible (uncertain) information in pervasive computing environments modelled by PNs. The PPN methodology has been formally described using matrix equations, and an algorithmic description has been provided as pseudocode to ease implementation. Three illustrative examples have been provided to help the reader to easily conceptualise the PPN procedure, and an application example has been used to demonstrate some of the challenges faced in a real-world application of PPNs.

The main conclusions of this work are as follows:

1. PPNs act as a hybrid system combining symbolic items (classical tokens that can represent external factors like resource availability, data arrival, etc.), and numerical values which represent uncertain knowledge about the system state through states of information;
2. We have seen how the information about the system state evolves and, in general, changes through information flow dynamics based on conjunction and disjunction of states of information;
3. The examples revealed that PPNs are capable of modelling cybersystems of different nature, since they can receive, store, exchange, and process information so as to use it for control;
4. The difficulty in the analytical evaluation of the conjunction of states of information has been highlighted as main drawback of PPNs. However, particle methods have been proposed to alleviate this difficulty with a feasible computational cost, although they come with a price of information loss due to the approximation by particles. The computational cost can be exacerbated in the presence of highly-dimensional state variables (say $d > 10$);
5. Building on this work, a future research direction is to formally explore further structural aspects of the resulting hybrid system, as well as to investigate efficient implementations to handle highly-dimensional spaces. Moreover, another future direction of research would be to investigate the adaptiveness of PPNs, i.e. their potential use for learning from data about the system state.

Acknowledgment

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/M023028/1], and also by the Lloyd's Register Foundation (LRF), a charitable foundation in the U.K. helping to protect life and property by supporting engineering-related education, public engagement, and the application of research. John Andrews is the LRF Director of the Resilience Engineering Research Group and also the Network Rail Professor of Infrastructure Asset Management at the University of Nottingham. The authors gratefully acknowledge the support of these organisations.

Appendix A. Proof of Theorem 1

Let us denote by $I_{k+1}^{p_j}$ the Shannon's information content that arises at place $p_j^{(\mathcal{N})} \in {}^\bullet t_i$ at $k+1$, after transition t_i is fired at time k. By definition of Shannon's information content [20,50]:

$$I_{k+1}^{p_j} = \int a_{ij}^{-} \left(f^{p_j} \wedge f^{t_i} \right) \log \frac{a_{ij}^{-} f^{p_j} \wedge f^{t_i}}{\mu} d\mathbf{x}_k \tag{30}$$

where $f^{p_j} \wedge f^{t_i}$ is assumed to be normalised. By Eq. (5):

$$\int a_{ij}^{-}\left(f^{p_j}\wedge f^{t_i}\right)\log\frac{a_{ij}^{-}f^{p_j}\wedge f^{t_i}}{\mu}d\mathbf{x}_k=\int a_{ij}^{-}\frac{f^{p_j}f^{t_i}}{\alpha_{ij}\mu}\log\frac{a_{ij}^{-}\frac{f^{p_j}f^{t_i}}{\alpha_{ij}\mu}}{\mu}d\mathbf{x}_k=$$
$$\int a_{ij}^{-}\frac{f^{t_i}f^{p_j}}{\alpha_{ij}\mu}\left(\log\frac{a_{ij}^{-}}{\alpha_{ij}\mu}+\log\frac{f^{t_i}f^{p_j}}{\mu}\right)d\mathbf{x}_k=$$
$$a_{ij}^{-}\left[\log\frac{a_{ij}^{-}}{\alpha_{ij}\mu}+\log\frac{1}{\mu}+\int\frac{f^{p_j}f^{t_i}}{\alpha_{ij}\mu}\log\left(f^{t_i}\right)d\mathbf{x}_k+\int\frac{f^{p_j}f^{t_i}}{\alpha_{ij}\mu}\log\left(f^{p_j}\right)d\mathbf{x}_k\right] \tag{31}$$

where we made use of the normalisation condition of $f^{p_j}\wedge f^{t_i}$, i.e. $\int\frac{f^{p_j}f^{t_i}}{\alpha_{ij}\mu}d\mathbf{x}_k=1$. Observe from the last equation that:

$$\int\frac{f^{p_j}f^{t_i}}{\alpha_{ij}\mu}\log\left(f^{t_i}\right)d\mathbf{x}_k=\mathbb{E}_{\text{conj}}\left[\log\left(f^{t_i}\right)\right], \tag{32a}$$

$$\int\frac{f^{p_j}f^{t_i}}{\alpha_{ij}\mu}\log\left(f^{p_j}\right)d\mathbf{x}_k=\mathbb{E}_{\text{conj}}\left[\log\left(f^{p_j}\right)\right], \tag{32b}$$

where $\mathbb{E}_{\text{conj}}[\cdot]$ denotes the expectation with respect to the probability density function given by the conjunction $f^{t_i}\wedge f^{p_j}$. Henceforth, Eq. (31) can be rewritten as:

$$I_{k+1}^{p_j}=a_{ij}^{-}\left[\log\frac{a_{ij}^{-}}{\alpha_{ij}\mu^2}+\mathbb{E}_{\text{conj}}\left[\log\left(f^{p_j}\right)\right]+\mathbb{E}_{\text{conj}}\left[\log\left(f^{t_i}\right)\right]\right] \tag{33}$$

By Jensen's inequality,

$$I_{k+1}^{p_j}\leqslant a_{ij}^{-}\left[\log\frac{a_{ij}^{-}}{\alpha_{ij}\mu^2}+\log\left(\mathbb{E}_{\text{conj}}\left[f^{p_j}\right]\right)+\log\left(\mathbb{E}_{\text{conj}}\left[f^{t_i}\right]\right)\right] \tag{34}$$

since the logarithm is concave. In the last equation, the logarithm of the expectation of f^{p_j} can be expressed as:

$$\log\left(\mathbb{E}_{\text{conj}}\left[f^{p_j}\right]\right)=\log\int f^{p_j}\frac{f^{t_i}f^{p_j}}{\alpha_{ij}\mu}d\mathbf{x}_k=\log\left(\frac{1}{\alpha_{ij}\mu}\underbrace{\int f^{p_j}f^{p_j}f^{t_i}d\mathbf{x}_k}_{\mathbb{E}_{f^{t_i}}\left[\left(f^{p_j}\right)^2\right]}\right)$$
$$=\log\frac{1}{\alpha_{ij}\mu}+\log\mathbb{E}_{f^{t_i}}\left[\left(f^{p_j}\right)^2\right] \tag{35}$$

In an analogue manner, $\log\left(\mathbb{E}_{\text{conj}}\left[f^{t_i}\right]\right)=\log\frac{1}{\alpha_{ij}\mu}+\log\mathbb{E}_{f^{p_j}}\left[\left(f^{t_i}\right)^2\right]$, so that Eq. (34) can be rewritten as:

$$I_{k+1}^{p_j}\leqslant a_{ij}^{-}\left[\log\frac{a_{ij}^{-}}{\mu^4\alpha_{ij}^3}+\log\mathbb{E}_{f^{p_j}}\left[\left(f^{t_i}\right)^2\right]+\log\mathbb{E}_{f^{t_i}}\left[\left(f^{p_j}\right)^2\right]\right] \tag{36}$$
$$=\log\left[\frac{a_{ij}^{-}}{\mu^4\alpha_{ij}^3}\mathbb{E}_{f^{t_i}}\left[\left(f^{p_j}\right)^2\right]\mathbb{E}_{f^{p_j}}\left[\left(f^{t_i}\right)^2\right]\right]^{a_{ij}^{-}}$$

Finally, we extend the last expression to the places $p_1,\ldots,p_m\in{}^{\bullet}t_i$ to consider the joint information from the pre-set of t_i denoted by ${}^{\bullet}I_{t_i}$, as follows:

$${}^{\bullet}I_{t_i}=\sum_{j=1}^{m}I_{k+1}^{p_j}\leqslant\log\left[\prod_{j=1}^{m}\left(\frac{a_{ij}^{-}}{\mu^4\alpha_{ij}^3}\mathbb{E}_{f^{t_i}}\left[\left(f^{p_j}\right)^2\right]\mathbb{E}_{f^{p_j}}\left[\left(f^{t_i}\right)^2\right]\right)^{a_{ij}^{-}}\right]=^{\bullet}I_{\max.t_i} \tag{37}$$

as we wanted to demonstrate.

Appendix B. Proof of Theorem 2

The content of information that arises in place $p_j^{(\mathcal{N})}\in t_i^{\bullet}$ after the transition t_i is fired, is expressed as [20,50]:

$$I_{k+1}^{p_j}=\int b_{ij}\left(f^{p_j}+a_{ij}^{+}f^{\text{conj}.t_i}\right)\log\left(\frac{b_{ij}}{\mu}\left(f^{p_j}+a_{ij}^{+}f^{\text{conj}.t_i}\right)\right)d\mathbf{x}_k$$

The last equation can be split into a sum of integrals as:

$$I_{k+1}^{p_j}=b_{ij}\left(\int f^{p_j}\log\left(\frac{b_{ij}}{\mu}(f^{p_j}+a_{ij}^{+}f^{\text{conj}.t_i})\right)d\mathbf{x}_k+a_{ij}^{+}\int f^{\text{conj}.t_i}\log\left(\frac{b_{ij}}{\mu}(f^{p_j}+a_{ij}^{+}f^{\text{conj}.t_i})\right)d\mathbf{x}_k\right)$$

After certain manipulations, the last equation can be expressed as:

$$I_{k+1}^{p_j} = b_{ij}\int f^{p_j}\log\left(f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right)d\mathbf{x}_k + \underbrace{a_{ij}^{+}b_{ij}}_{1-b_{ij}}\int f^{\mathrm{conj},t_i}\log\left(f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right)d\mathbf{x}_k + \underbrace{(a_{ij}^{+}+1)b_{ij}}_{1}\log\frac{b_{ij}}{\mu} =$$

$$\log\frac{b_{ij}}{\mu} + b_{ij}\mathbb{E}_{f^{p_j}}\left[\log\left(f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right)\right] + (1-b_{ij})\mathbb{E}_{f^{\mathrm{conj},t_i}}\left[\log\left(f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right)\right]$$

By Jensen's inequality,

$$I_{k+1}^{p_j} \leqslant \log\frac{b_{ij}}{\mu} + b_{ij}\log\mathbb{E}_{f^{p_j}}\left[f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right] + (1-b_{ij})\log\mathbb{E}_{f^{\mathrm{conj},t_i}}\left[f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right]$$

which finally leads to:

$$I_{k+1}^{p_j} \leqslant \log\left[\frac{b_{ij}}{\mu}\left(\mathbb{E}_{f^{p_j}}\left[f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right]\right)^{b_{ij}}\left(\mathbb{E}_{f^{\mathrm{conj},t_i}}\left[f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right]\right)^{1-b_{ij}}\right] \tag{38}$$

By extending to the m places that belong to the post-set of t_i and after some manipulation, then the joint information $I_{t_i}^{\bullet}$ from the post-set of t_i can be finally expressed as:

$$I_{t_i}^{\bullet} = \sum_{j=1}^{m} I_{k+1}^{p_j} \leqslant \log\left[\prod_{j=1}^{m}\frac{b_{ij}}{\mu}\left(\mathbb{E}_{f^{p_j}}\left[f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right]\right)^{b_{ij}}\left(\mathbb{E}_{f^{\mathrm{conj},t_i}}\left[f^{p_j}+a_{ij}^{+}f^{\mathrm{conj},t_i}\right]\right)^{1-b_{ij}}\right] = I_{\max,t_i}^{\bullet} \tag{39}$$

Appendix C. Proof of Lemma 2

We need to demonstrate that each probability density from $\mathbf{M}_{k+1}^{(\mathcal{N})}$ integrates to unity. By taking integrals in Eq. (15):

$$\int \mathbf{M}_{k+1}^{(\mathcal{N})}d\mathbf{x}_{k+1} =$$

$$\left[\int \mathbf{M}_{k}^{(\mathcal{N})}\circ\boldsymbol{\gamma}_k d\mathbf{x}_k + \int\left(\sum_{i=1}^{n_t^{(\mathcal{N})}}(\mathbf{a}_i^{+})^T\otimes\mathbf{c}_i\right)\mathbf{v}_k d\mathbf{x}_k + \int\left((\mathbf{A}^{-})^T\circ\mathbf{B}\right)\mathbf{v}_k d\mathbf{x}_k\right]\circ\boldsymbol{\beta}_k \tag{40}$$

First, note that the integral

$$\int \mathbf{M}_{k}^{(\mathcal{N})}\circ\boldsymbol{\gamma}_k d\mathbf{x}_k = \begin{pmatrix}\gamma_k^{(1)}\\ \vdots \\ \gamma_k^{(n_p^{(\mathcal{N})})}\end{pmatrix} \tag{41}$$

by the assumption that the density functions from $\mathbf{M}_{k}^{(\mathcal{N})}$ are normalised. Next, the second term of the rightmost part of Eq. (40) can be evaluated as follows:

$$\int\left(\sum_{i=1}^{n_t^{(\mathcal{N})}}(\mathbf{a}_i^{+})^T\otimes\mathbf{c}_i\right)\mathbf{v}_k d\mathbf{x}_k = \tag{42}$$

$$\int\begin{pmatrix} a_{11}^{+}v_{1,k}\left(f^{t_1}\wedge f^{\bullet t_1}\right) & \cdots & a_{n_t^{(\mathcal{N})}1}^{+}v_{n_t^{(\mathcal{N})},k}\left(f^{t_{n_t^{(\mathcal{N})}}}\wedge f^{\bullet t_{n_t^{(\mathcal{N})}}}\right)\\ & \vdots & \\ a_{1n_p^{(\mathcal{N})}}^{+}v_{1,k}\left(f^{t_1}\wedge f^{\bullet t_1}\right) & \cdots & a_{n_t^{(\mathcal{N})}n_p^{(\mathcal{N})}}^{+}v_{n_t^{(\mathcal{N})},k}\left(f^{t_{n_t^{(\mathcal{N})}}}\wedge f^{\bullet t_{n_t^{(\mathcal{N})}}}\right)\end{pmatrix}d\mathbf{x}_k = \begin{pmatrix}\sum_{i=1}^{n_t^{(\mathcal{N})}}a_{i1}^{+}v_{i,k}\\ \vdots \\ \sum_{i=1}^{n_t^{(\mathcal{N})}}a_{in_p^{(\mathcal{N})}}^{+}v_{i,k}\end{pmatrix} \tag{43}$$

We can proceed analogously to evaluate the next integral, so that:

$$\int\left((\mathbf{A}^{-})^T\circ\mathbf{B}\right)\cdot\mathbf{v}_k d\mathbf{x}_k = \begin{pmatrix}\sum_{i=1}^{n_t^{(\mathcal{N})}}a_{i1}^{-}v_{i,k}\\ \vdots \\ \sum_{i=1}^{n_t^{(\mathcal{N})}}a_{in_p^{(\mathcal{N})}}^{-}v_{i,k}\end{pmatrix} \tag{44}$$

Finally, by making substitutions of Eqs. (41), (42), and (44) into (40), then:

$$\int \mathbf{M}_{k+1}^{(\mathcal{N})} d\mathbf{x}_{k+1} = \begin{pmatrix} \gamma_k^{(1)} + \sum_{i=1}^{n_t^{(\mathcal{N})}} \left(a_{i1}^- + a_{i1}^+\right) v_{i,k} \\ \vdots \\ \gamma_k^{(n_p^{(\mathcal{N})})} + \sum_{i=1}^{n_t^{(\mathcal{N})}} \left(a_{in_p^{(\mathcal{N})}}^- + a_{in_p^{(\mathcal{N})}}^+\right) v_{i,k} \end{pmatrix} \circ \begin{pmatrix} \frac{1}{\gamma_k^{(1)} + \sum_{i=1}^{n_t^{(\mathcal{N})}} \left(a_{i1}^- + a_{i1}^+\right) v_{i,k}} \\ \vdots \\ \frac{1}{\gamma_k^{(n_p^{(\mathcal{N})})} + \sum_{i=1}^{n_t^{(\mathcal{N})}} \left(a_{in_p^{(\mathcal{N})}}^- + a_{in_p^{(\mathcal{N})}}^+\right) v_{i,k}} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

as we wanted to demonstrate.

References

[1] R.E. Neapolitan, Probabilistic Reasoning in Expert Systems: Theory and Algorithms, Wiley, 1989.
[2] P. Krause, D. Clark, Representing Uncertain Knowledge: An Artificial Intelligence Approach, Springer Science & Business Media, 2012.
[3] A. Ramírez-Noriega, R. Juárez-Ramírez, S. Jiménez, Y. Martínez-Ramírez, Knowledge representation in intelligent tutoring system, in: Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, Springer, 2016, pp. 12–21.
[4] P. Larrañaga, S. Moral, Probabilistic graphical models in artificial intelligence, Appl. Soft Comput. 11 (2) (2011) 1511–1528.
[5] C.A. Petri, Kommunikation mit Automaten, Institut für Instrumentelle Mathematik an der Universität Bonn, 1962 (Ph.D. thesis).
[6] T. Murata, Petri nets: properties, analysis and applications, Proc. IEEE 77 (4) (1989) 541–580.
[7] M. Silva, Half a century after Carl Adam Petri's Ph. D. thesis: a perspective on the field, Ann. Rev. Control 37 (2) (2013) 191–219.
[8] R. Zurawski, M. Zhou, Petri nets and industrial applications: a tutorial, IEEE Trans. Ind. Electron. 41 (6) (1994) 567–583.
[9] S.-M. Chen, J.-S. Ke, J.-F. Chang, Knowledge representation using fuzzy Petri nets, IEEE Trans. Knowl. Data Eng. 2 (3) (1990) 311–319.
[10] H.-C. Liu, L. Liu, Q.-L. Lin, N. Liu, Knowledge acquisition and representation using fuzzy evidential reasoning and dynamic adaptive fuzzy Petri nets, IEEE Trans. Cybern. 43 (3) (2013) 1059–1072.
[11] C.G. Looney, Fuzzy Petri nets for rule-based decision making, IEEE Trans. Syst. Man Cybern. 18 (1) (1988) 178–183.
[12] H.-C. Liu, J.-X. You, X.-Y. You, Q. Su, Linguistic reasoning Petri nets for knowledge representation and reasoning, IEEE Trans. Syst. Man Cybern. Syst. 46 (4) (2016) 499–511.
[13] H.-C. Liu, J.-X. You, X.-Y. You, Q. Su, Fuzzy Petri nets using intuitionistic fuzzy sets and ordered weighted averaging operators, IEEE Trans. Cybern. 46 (8) (2016) 1839–1850.
[14] J. Cardoso, R. Valette, D. Dubois, Fuzzy Petri nets: an overview, in: Proceedings of the Thirteenth Word Congress of IFAC, 1, 1996, pp. 443–448.
[15] H.-C. Liu, Q.-L. Lin, L.-X. Mao, Z.-Y. Zhang, Dynamic adaptive fuzzy Petri nets for knowledge representation and reasoning, IEEE Trans. Syst. Man Cybern. Syst. 43 (6) (2013) 1399–1410.
[16] K.-Q. Zhou, A.M. Zain, Fuzzy Petri nets and industrial applications: a review, Artif. Intell. Rev. 45 (4) (2016) 405–446.
[17] H.-C. Liu, J.-X. You, Z. Li, G. Tian, Fuzzy Petri nets for knowledge representation and reasoning: a literature review, Eng. Appl. Artif. Intell. 60 (2017) 45–56.
[18] J. Cardoso, R. Valette, D. Dubois, Possibilistic Petri nets, IEEE Trans. Syst. Man Cybern. Part B Cybern. 29 (5) (1999) 573–582.
[19] J. Lee, F. Kevin, W. Chiang, Modeling uncertainty reasoning with Possibilistic Petri nets, IEEE Trans. Syst. Man Cybern. Part B Cybern. 33 (2) (2003) 214–224.
[20] A. Tarantola, B. Valette, Inverse problems = quest for information, J. Geophys. 50 (3) (1982) 159–170.
[21] G. Rus, J. Chiachío, M. Chiachío, Logical inference for inverse problems, Invers. Probl. Sci. Eng. 24 (3) (2016) 448–464.
[22] R. David, H. Alla, Continuous Petri nets, in: Procceedings of the Eight European Workshop on Application and Theory of Petri Nets, Zaragoza, Spain, 1988, pp. 275–294.
[23] M. Silva, J. Colom, On the structural computation of synchronic invariants in P/T nets, in: Procceedings of the Eight European Workshop on Application and Theory of Petri Nets, Zaragoza, Spain, 1998, pp. 237–258.
[24] R. David, H. Alla, Autonomous and timed continuous Petri nets, in: G. Rozenberg (Ed.), Advances in Petri Nets 1988, Lecture Notes in Computer Science, 674, Springer, New York, 1993, pp. 71–90.
[25] M. Silva, J. Colom, On the computation of structural synchronic invariants in P/T nets, in: G. Rozenberg (Ed.), Advances in Petri Nets 1988, Lecture Notes in Computer Science, 340, Springer, New York, 1998, pp. 387–417.
[26] L. Ghomri, H. Alla, Modeling and analysis using hybrid Petri nets, Nonlinear Anal. Hybrid Syst. 1 (2) (2007) 141–153.
[27] J. Júlvez, S. Di Cairano, A. Bemporad, C. Mahulea, Event-driven model predictive control of timed hybrid Petri nets, Int. J. Robust Nonlinear Control 24 (12) (2014) 1724–1742.
[28] X. Allamigeon, V. Boeuf, S. Gaubert, Stationary solutions of discrete and continuous Petri nets with priorities, Perform. Evaluat. 113 (2017) 1–12.
[29] M. Silva, Individuals, populations and fluid approximations: a Petri net based perspective, Nonlinear Anal. Hybrid Syst. 22 (2016) 72–97.
[30] K.S. Trivedi, V.G. Kulkarni, FSPNs: Fluid Stochastic Petri nets, Springer, Berlin, Heidelberg, pp. 24–31.
[31] C.R. Vazquez, M. Silva, Stochastic hybrid approximations of Markovian Petri nets, IEEE Trans. Syst. Man Cybern. Syst. 45 (9) (2015) 1231–1244.
[32] C. Lesire, C. Tessier, Particle Petri nets for aircraft procedure monitoring under uncertainty, in: G. Ciardo (Ed.), Proceedings of the Twenty-Sixth International Conference on Applications and Theory of Petri Nets, ICATPN, Springer, Miami, USA, 2005, pp. 329–348. June 20-25.
[33] L. Zouaghi, A. Alexopoulos, A. Wagner, E. Badreddin, Modified particle Petri nets for hybrid dynamical systems monitoring under environmental uncertainties, in: Proceedings of the IEEE/SICE International Symposium on System Integration (SII), IEEE, 2011, pp. 497–502.
[34] Q. Gaudel, E. Chanthery, P. Ribot, Hybrid particle Petri nets for systems health monitoring under uncertainty, Int. J. Prognost. Health Manag. 6 (Special Issue Uncertainty in PHM) (2015) 22.
[35] M. Chiachío, J. Chiachío, D. Prescott, J. Andrews, An information theoretic approach for knowledge representation using Petri nets, in: Proceedings of the Future Technologies Conference, IEEE, San Francisco, 2016, pp. 165–172. 6–7 December.
[36] M.K. Molloy, Performance analysis using stochastic Petri nets, IEEE Trans. Comput. 31 (9) (1982) 913–917.
[37] M.A. Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, A. Cumani, The effect of execution policies on the semantics and analysis of stochastic petri nets, IEEE Trans. Softw. Eng. 15 (7) (1989) 832–846.
[38] E. Jaynes, G. Bretthorst, Probability Theory: The Logic of Science, Cambridge University Press, 2003.
[39] E. Jaynes, Papers on Probability, Statistics and Statistical Physics, Kluwer Academic Publishers, 1983.
[40] K. Mosegaard, A. Tarantola, Probabilistic approach to inverse problems, in: International Handbook of Earthquake and Engineering Seismology, Part A, in: International Geophysics, 81, Academics Press Ltd, 2002, pp. 237–265.
[41] M. Arumlampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.
[42] A. Tarantola, K. Mosegaard, Mapping of Probabilities, Theory for the Interpretation of Uncertain Physical Measurements, Cambridge University Press, 2007.
[43] L. Devroye, Non-Uniform Random Variate Generation, Springer-Verlag, New York, 1986.
[44] R.A. Beezer, A First Course in Linear Algebra, 1.08 ed., Robert A. Beezer, 2007.

[45] B. Fryc, K. Pancerz, J.F. Peters, Z. Suraj, On fuzzy reasoning using matrix representation of extended fuzzy petri nets, Fundam. Inf. 60 (1–4) (2004) 143–157.
[46] Z. Suraj, Matrix representation of parameterised fuzzy Petri nets., in: D. Ciucci, M. Inuiguchi, Y. Yao, D. Slezak, G. Wang (Eds.), Proceedings of the RSFDGrC, Lecture Notes in Computer Science, 8170, Springer-Verlag, Berlin Heidelberg, 2013, pp. 200–207. Halifax, Canada, October 11–14.
[47] M.-G. Chun, Z. Bien, Fuzzy petri net representation and reasoning methods for rule-based decision making systems, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 76 (6) (1993) 974–983.
[48] W. Pedrycz, H. Camargo, Fuzzy timed Petri nets, Fuzzy Sets Syst. 140 (2) (2003) 301–330.
[49] D.D.H. Prade, D. Dubois, Possibility Theory: An Approach to Computerized Processing of Uncertainty, Plenum Press, 1988.
[50] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (0) (1948) 379–423.